

Департамент информатизации Тюменской области
Государственное автономное учреждение дополнительного образования
Тюменской области

«Региональный информационно-образовательный центр»

СОГЛАСОВАНО

УТВЕРЖДАЮ

Директор
Департамента информатизации
Тюменской области
_____ М.В.Рудзевич

Директор
ГАУ ДО ТО «РИО-Центр»
_____ Т.А. Беляева

«_____» _____ 2021г

«_____» _____ 2021г

УЧЕБНАЯ ПРОГРАММА
«Программирование на языке Python»

Трудоемкость программы – 42 академических часа

Форма обучения – очная

Режим занятий – 3 академических часа в день

Начальные навыки: Базовые навыки работы на персональном компьютере

Цель обучения:

Познакомить школьников с основами алгоритмики, сформировать у школьников комплекс знаний и умений, необходимый для разработки программ и игр на языке Python.

Компетенции на выходе:

- Среда разработки на языке Python, знакомство с интерфейсом и функционалом, инсталляции среды;
- Типы данных;
- Переменные;
- Основы синтаксиса языка Python;
- Ветвления, принятие решений и циклы;
- Работа с массивами;
- Функции: создание и вызов функций, передача аргументов в функцию, получение данных из функции.

УЧЕБНЫЙ ПЛАН

№ п/ п	Наименование тем	Всего часов	в том числе:		Формы контроля
			Теория	Практи ка	
Приветственный кейс					
1.	Приветственный кейс	7	2	5	
2.	Вводный кейс	8	3	5	
3.	Основной кейс	12	3	9	
4.	Продвинутый кейс	15	5	10	
Итого часов:		42	14	29	

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Наименование тем	Всег о часо в	в том числе:		Формы контроля
			Теория	Практи ка	
1. Приветственный кейс(выбор одного кейса приветственного уровня из базы кейсов)					
1.1.	Кейс « Just do it! »	7	2	5	
1.2.	Кейс « Let’s travel »				
2. Вводный кейс(выбор одного кейса вводного уровня из базы кейсов)					
2.1.	Кейс « Угадай число »	8	3	5	
2.2.	Кейс « Навстречу приключениям »				
3. Основной кейс(выбор одного кейса основного уровня из базы кейсов)					
3.1.	Кейс « Черепашья графика »	12	3	9	
3.2.	Кейс « Черепашка, математика и случайные числа »				
4. Продвинутый кейс(выбор одного кейса продвинутого уровня из базы кейсов)					
3.1.	Кейс « Поле чудес »	15	5	10	
3.2.	Кейс « Сам себе чертежник »				
Итого часов:		42	14	29	

СОДЕРЖАНИЕ ПРОГРАММЫ

Сводная информация о темах, раскрытых в рамках курса

Наименование темы	Кейсы, в которых раскрывается содержание темы
Среда разработки на языке Python, знакомство с интерфейсом и функционалом, инсталляции среды	Кейс № 1 «Just do it» Кейс № 2 «Let's travel»
Типы данных	Кейс № 1 «Just do it» Кейс № 2 «Let's travel» Кейс № 3 «Угадай число» Кейс № 4 «Навстречу приключениям»
Переменные	Кейс № 1 «Just do it» Кейс № 2 «Let's travel» Кейс № 3 «Угадай число» Кейс № 4 «Навстречу приключениям»
Основы синтаксиса языка Python	Кейс № 1 «Just do it» Кейс № 2 «Let's travel» Кейс № 3 «Угадай число» Кейс № 4 «Навстречу приключениям»
Ветвления, принятие решений и циклы	Кейс № 3 «Угадай число» Кейс № 4 «Навстречу приключениям»
Работа с массивами	Кейс № 7 «Поле чудес» Кейс № 8 «Сам себе чертежник»
Функции: создание и вызов функций, передача аргументов в функцию, получение данных из функции	Кейс № 5 «Черепашья графика» Кейс № 6 «Черепашка, математика и случайные числа» Кейс № 7 «Поле чудес» Кейс № 8 «Сам себе чертежник»

База кейсов

Кейс № 1

«Just do it!»

Текст кейса

В этом кейсе вам предстоит познакомиться с языком программирования Python и применить его для решения практической географической задачи. Из курса географии вам известно, что при увеличении высоты температура воздуха понижается, а при уменьшении – повышается. Причем есть определенная закономерность: при изменении высоты на 1000 метров (1 км) температура воздуха изменяется на 6°C.

Используя ресурсы сети Интернет, определите температуру у подножия высочайших вершин мира и их высоту. А определить температуру на пике вершины вам поможет язык программирования Python.

Описание продукта

Высотный калькулятор температуры: по известным данным о температуре на определенной высоте он будет рассчитывать температуру на другой высоте в той же точке.

Исходные данные вводятся пользователю с клавиатуры. Результаты выводятся на экран.

Для ввода и вывода информации используйте поясняющие надписи. Уведомляйте пользователя о старте и завершении программы.

Дополнительно: создать калькулятор, который будет определять разницу высот по двум замерам температуры.

Категория кейса: приветственный.

Вопросы к кейсу:

1. Что такое программирование?
2. Для чего придумали языки программирования?
3. Кто и когда придумал язык программирования Python и зачем, если до этого уже существовало много языков программирования?
4. Какие фишки есть у языка программирования Python?
5. Где сейчас используется язык программирования Python?

Количество учебных часов/занятий, на которые рассчитан кейс:

7 ак. ч / 3,5 занятия

Учебно-тематическое планирование (занятие — 2 часа):

Занятие 1	
Цель: Познакомиться, презентовать курс, сформировать у обучающихся понимание, что такое программирование и зачем придумали языки программирования. Установить Python, познакомить обучающихся с работой в интерактивном режиме (REPL, от англ. read-eval-print loop — «цикл „чтение — вычисление — вывод“»), показать, какие операции могут быть выполнены над целыми и действительными числами, а также несколько операций над строками.	
Что делает педагог:	Что делают дети:
1. Приветствие. Представление педагога. Игра на знакомство. Предлагает учащимся сесть в круг. Берет несколько салфеток из пачки, пускает пачку по кругу и предлагает обучающимся сделать то же самое - каждому взять несколько салфеток из пачки. Затем рассказывает столько фактов о себе, сколько салфеток у него в рука, и предлагает учащимся сделать то же самое.	Садятся в круг, берут несколько салфеток из пачки, рассказывают столько фактов о себе, сколько салфеток в пачке.
2. Представление курса и обмен ожиданиями.	Рассказывают, как они попали на этот курс, и какие у них цели на этот курс.
3. Делит участников на команды по 3-4 человека: по желанию, жеребьевкой	Делятся на команды предложенным педагогом способом.
3. Раскладывает вниз рубашкой заранее подготовленные карточки с вопросами к кейсу: - Что такое программирование? - Для чего придумали языки программирования? - Кто и когда придумал язык программирования Python и зачем, если до этого уже существовало много языков программирования? - Какие фишки есть у языка программирования Python? - Где сейчас используется язык программирования Python? Резюмирует ответы команд участников и добавляет необходимую информацию.	Команды участников вытягивают вопрос, на который им предстоит искать ответ, и ищут на него ответ, используя ресурсы, предложенные педагогом, или другие ресурсы сети Internet (10–15 минут). Затем команды участников презентуют свои ответы.
2. Устанавливает Python в реальном времени или показывает и рассказывает обучающимся, как это сделать: https://www.python.org/downloads/ (для экономии времени можно скачать дистрибутив Python заранее). Проверяет или показывает, как проверить, что Python установлен на компьютере (меню «Пуск» → папка «Python 3.9»)	Повторяют действия за педагогом на своих персональных компьютерах.
3. Запускает интерпретатор Python (Python 3.9 (64-bit)), показывает, что он может работать как «умный» калькулятор на ряде примеров. Сложение и вычитание: – Python умеет складывать и вычитать, в том числе в	Повторяют действия за педагогом на своих персональных компьютерах. Повторяют действия за педагогом на своих

ответе могут получиться отрицательные числа; - если нужно сложить или вычесть отрицательное число, то скобки можно писать или не писать. Предлагает обучающимся придумать свои примеры на сложение и вычитание: - чтобы в конце получилось отрицательное число - чтобы участвовали отрицательные числа. Резюмирует.	персональных компьютерах. Придумывают свои примеры на сложение и вычитание, чтобы в конце получилось отрицательное число. Придумывают свои примеры на сложение и вычитание с отрицательными числами.
4. Показывает, что интерпретатор Python может работать как «умный» калькулятор на ряде примеров (продолжение). Умножение и деление: - Python умеет умножать и делить, причем в ответе могут получиться дробные числа; - в качестве разделителя в языке программирования Python используется точка. Определяет понятия оператора и операнда. Предлагает обучающимся придумать свои примеры такие, чтобы: – в ответе получилось число с дробной частью; – числа с дробной частью использовались в сложении, вычитании, умножении, делении; - проверить, что происходит при делении на ноль. Резюмирует.	Придумывают свои примеры умножение и деление, в том числе такие, чтобы - в ответе получилось число с дробной частью; - числа с дробной частью использовались в сложении, вычитании, умножении, делении.
5. Показывает, что интерпретатор Python может работать как «умный» калькулятор на ряде примеров (продолжение). Скобки. Предлагает обучающимся расставить скобки, чтобы получились верные равенства: $9+9+9/9+9+9+9 = 10$ $99-9:9*9+9:9=100$ Предлагает обучающимся придумать свои примеры, такие, где от расстановки скобок зависит результат (обучающимся нужно вывести несколько вариантов: со скобками и без, с разными способами расстановки скобок).	Пробуют расставить скобки в примерах преподавателя для получения верного равенства. Придумывают свои примеры со скобками такие, где от расстановки скобок зависит результат.
6. Предлагает обучающимся вспомнить про приоритет математических действий и назвать их в порядке убывания приоритета. Фиксирует операторы по убыванию приоритета на доске или флипчарте (можно в виде таблицы со столбцами "Приоритет", "Оператор", "Математическая операция") () скобки ** возведение в степень +x, -x, ~x унарные плюс, минус *, /, //, % умножение, деление, частное от деления с остатком, взятие остатка +, - сложение и вычитание	Обучающиеся вспоминают математические операции и их приоритет.
7. Показывает, что интерпретатор Python может работать как «умный» калькулятор на ряде	Придумывают свои примеры на возведение в степень и

примеров (продолжение). Возведение в степень и извлечения корня. Обращает внимание, что при извлечении корня нужно использовать скобки, так как оператор возведения в степень имеет больший приоритет, чем другие операторы.	извлечение корня, в том числе «неквадратного»
7. Говорит о том, в программах на самых разных языках программирования кроме чисел, часто используются строки. Например, чтобы вывести пользователю приглашение для ввода или прокомментировать вывод. - В языке программирования Python для обозначения строк используются одинарные (апострофы) или двойные кавычки. - Расстояние между словами отражается пробелом. - Знак препинания - тоже часть строки. Строка может состоять только из одного знака препинания или только из нескольких знаков препинания. - Строки могут содержать любые печатные символы, которые мы вводим с клавиатуры (стоит аккуратнее быть с кавычками внутри строк и обратным слешем \ - об этом будет рассказано позже). Предлагает обучающимся сформировать определенные строки: - строку, содержащую их имя (с заглавной буквы) и обратить внимание, сохраняет ли Python регистр; - строку, содержащую их имя, отчество и фамилию с пробелами между ними; - строки, содержащие только русские, только английские, и те и те буквы; - пустую строку; - строку из одного пробела; - строку из нескольких пробелов; - строку из одного знака препинания; - строку из символов не букв и не цифр; - строку, симметричную относительно своего центра (например, "%\$шалаш\$%", можно напомнить о существовании таких слов и фраз как палиндромы); - строку с определенным порядком (например, "АббААбббАААбббб"). Предлагает обучающимся применить к строкам операцию сложения (конкатенации) и дублирования ("умножения"): - составить предложение из отдельных слов и знаков препинания; - продублировать строки (например, если продублировать строки "МЫШКА", "БАНКА", "КУПИЛ", "СЫРО" в получившейся строке можно увидеть другие слова - камыш, кабан, пилка, росы). Резюмирует.	Обучающиеся формируют строки: - строку, содержащую их имя (с заглавной буквы) и обратить внимание, сохраняет ли Python регистр; - строку, содержащую их имя, отчество и фамилию с пробелами между ними; - строки, содержащие только русские, только английские, и те и те буквы; - пустую строку; - строку из одного пробела; - строку из нескольких пробелов; - строку из одного знака препинания; - строку из символов не букв и не цифр; - строку, симметричную относительно своего центра (например, "%\$шалаш\$%", можно напомнить о существовании таких слов и фраз как палиндромы); - строку с определенным порядком (например, "АббААбббАААбббб").

8. Резюмирует, что было сделано на занятии.	Дополняют резюме преподавателя
9. Предлагает обучающимся высказать свои впечатления от занятия.	Высказывают свои впечатления от занятия
Занятие 2	
Цель: Познакомить обучающихся с интерактивной средой разработки IDLE. Объяснить, что такое переменные и типы данных и зачем они нужны. Рассказать о том, зачем и как сохранять программы, написанные на языке программирования Python в файл. Рассказать, как выводить, как запрашивать у пользователя информацию на примере задачи о расчете периметра и площади прямоугольника. Самостоятельная работа над кейсом (продолжение).	
Что делает педагог:	Что делают дети:
1. Напоминает, чему было посвящено предыдущее занятие. Озвучивает цель и план занятия: - продолжение знакомства друг с другом (игра «Посылка»); - продолжение знакомства с языком программирования Python; - знакомство с кейсом, при необходимости разделение на пары / команды; - самостоятельная работа над кейсом, консультации преподавателя; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой.	
2. Проводит игру на продолжение знакомства «Посылки» и помогает всем учащимся активно включиться в игру. Игра проводится в круге (лучше сидя, но можно и стоя). Для игры понадобится небольшой мячик. Преподаватель называет себя, называет по имени любого участника, кроме тех, кто рядом с ним, придумывает ему посылку на букву, с которой начинается его имя и кидает ему мяч. Далее посылку отправляет уже этот участник. Посылки каждый раз разные. На первую букву имени получателя. Отправлять посылку дважды одному участнику нельзя. Посылки не должны повторяться. Когда последний участник получит посылку, можно сделать возврат посылок по тому же маршруту. Обычно это интересно и проходит при активной поддержке участников, так как нужно вспомнить не только того, кто слал тебе посылку, но и что именно он тебе слал.	Участвуют в игре. При необходимости помогают другим участникам вспомнить имена.
2. Знакомит обучающихся с интерактивной средой разработки IDLE: - как запустить; - интерактивный режим.	Выполняют несколько действий в интерактивной среде разработки IDLE. Обсуждают неудобство использования интерактивного

	режима при программном решении сложных задач.
3. Дает определение встроенных функций языка Python и их параметров, приводит примеры: - функция len() - для определения длины строки; - функция print() - для вывода информации на экран, информация может быть текстовой и числовой (выведенные параметры разделяются пробелами, если нужен другой разделитель - используется именованный параметр sep). Предлагает обучающимся определить количество символов в словах: абракадабра, автоэлектростеклоподъемники, всплеск (7 букв, 1 слог) и др. Вывести на экран любую пословицу, используя одну строку или несколько строк.	Определяют количество символов в словах: абракадабра, автоэлектростеклоподъемники, всплеск (7 букв, 1 слог) и др. Выводят на экран любую пословицу, используя одну строку или несколько строк.
4. Подводит обучающихся к мысли о неудобстве интерактивного режима. Предлагает обсудить, будет ли удобен интерактивный режим при программном решении сложных задач. Например, необходимо вывести на экран четверостишие, используя функцию print() для каждой строки. Резюмирует: Будет удобнее, если разместить этот код в файле.	Пробуют вывести на экран четверостишие, используя функцию print() для каждой строки.
5. Демонстрирует, как создать новый файл с кодом программы на Python (с расширением *.py), как сохранить его, а затем заново открыть и запустить.	Повторяют действия за преподавателем.
6. Повторно предлагает обучающимся вывести на экран четверостишие, используя функцию print() для каждой строки, и разместить этот код в файле.	Создают файл *.py для вывода на экран четверостишия, используя функцию print() для каждой строки.
7. Подводит обучающихся к мысли о необходимости переменных. Озвучивает задачу: Автомобиль движется по дороге с определенной скоростью - 25 м/с. Определить, сколько км проедет автомобиль за 1 ч, 2ч, 5 ч, 7 ч, 10 ч, 12 ч и т.д. Предлагает увеличить или уменьшить скорость? Что теперь нужно делать с вычислениями? Везде заменять 25 на другую величину - неудобно. Лучше скорость где-то хранить а затем для вычислений просто подставлять. Озвучивает определение переменной и механизм создания и использования переменных в Python. Приводит примеры создания переменных разных типов.	Создают файл *.py для расчета пробега автомобиля.
8. Рассказывает, как можно менять тип и между какими типами это возможно.	Проверяют действие функций преобразования типов int(), float(), str(), bool()

9. Подводит обучающихся к мысли о том, что не всегда удобно задавать значения переменных сразу в программу. В ряде случаев значения должен вводить пользователь. Также в ряде случаев необходимо хранить информацию, которая будет использоваться позднее.	Предлагают свои варианты, как поступить в этом случае.
11. Дает определение переменной, рассказывает о том, что Python определяет тип данных переменной, исходя из значения, которое ей присвоено (на профессиональном языке - является языком с динамической типизацией).	Создают несколько переменных разных типов. Проверяют тип переменных функцией <code>type()</code> .
12. Рассказывает, как ввести значение с клавиатуры в переменную - с помощью функции <code>input()</code>. Как сделать приглашение для ввода. Значение какого типа вводится по умолчанию (строка - <code>str</code>), как изменить тип вводимого значения. Предлагает ввести 2 значения, один раз привести их к типу <code>int()</code> , а другой раз - к <code>str()</code> и сложить (результат будет разным).	Пробуют вводить значения в переменные с клавиатуры и изменять тип данных переменных. Вводят с клавиатуры 2 значения, од раз привести их к типу <code>int()</code> , а другой раз - к <code>str()</code> и сложить (результат будет разным).
13. Резюмирует, что умеют делать обучающиеся на языке программирования Python на данный момент.	Дополняют резюме преподавателя.
14. Озвучивает кейс для самостоятельной работы и форму презентации результатов по нему. При необходимости предлагает обучающимся разбиться на пары / команды.	При необходимости разбиваются на пары / команды.
15. Озвучивает необходимость составления плана работ и назначения ответственных (на листе бумаге, на флипчарте с помощью стикеров, в программном обеспечении Trello или другом).	Выбирают средство планирования, составляют план работ и назначают ответственных (при работе в паре / команде)
16. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости.	Самостоятельно работают над кейсом
17. Предлагает обучающимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.
Занятие 3	
Цель: Самостоятельная работа над кейсом (продолжение). Подготовка материалов для презентации результатов кейса	
Что делает педагог:	Что делают дети:
1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут способы

кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; В дальнейшем следит за таймингом.	преодоления возникших трудностей.
2. Наблюдает за продвижением участников / пар / команд, предупреждает возникновение конфликтов, консультирует по мере необходимости. Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на интересующую тему, или натолкнуть его на мысль, как получить результат на основе тех знаний, которые у него есть в данный момент.	Самостоятельно работают над кейсом.
3. Озвучивает необходимость перейти к подготовке презентации для защиты результатов проекта	Готовят презентацию для защиты результатов кейса.
4. Предлагает обучающимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут способы преодоления возникших трудностей.
Занятие 4 (1 час)	
Цель: Провести демонстрацию полученного за кейс результата, провести рефлекссию	
Что делает педагог:	Что делают дети:
1. Озвучивает цель занятия, определяет порядок демонстрации результатов кейса. Например, на выступление каждой команды не более 3-х минут, далее в течение 2-х минут вопросы.	
2. Устанавливает порядок, в котором обучающиеся будут демонстрировать свои результаты, или порядок может быть случайным: раскладывает на столе карточки с номером выступления	Каждый участник / команда берет карточку с номером выступления
3. Осуществляет модерацию презентаций результатов кейса участниками.	Участники / команда презентуют результаты кейса
4. Проводит рефлексию (лучше проводить, сидя в кругу). Задает вопросы обучающимся: 1. Расскажите, чему вы научились за этот кейс, какие навыки освоили. 2. Удалось ли применить полученные навыки еще где-то? (возможно, вы установили интерпретатор Python и интегрированную среду разработки IDLE на своем домашнем компьютере / ноутбуке и показали работу интерпретатора Python своим друзьям / родителям / братьям и сестрам). 3. Что в этом кейсе далось вам легко, а что было	Отвечают на вопросы преподавателя: 1. Чему научились за этот кейс, какие навыки освоили. 2. Удалось ли применить полученные навыки еще где-то? . 3. Что в этом кейсе далось легко, а что было трудным для освоения? Как справились с этими трудностями?

трудным для освоения? Как вы справились с этими трудностями?	
--	--

Минимально необходимый уровень входных компетенций: базовое владение ПК.

Предполагаемые результаты обучающихся, формируемые навыки:

- обучающиеся научатся взаимодействовать с интерпретатором Python, создавать и запускать простые программы на языке Python в файлах с расширением *.py в интегрированной среде разработки IDLE;
- освоят понятия «язык программирования», «среда программирования», «скрипт», «инструкция», «операция», «линейный алгоритм», «переменная», «тип данных», «встроенная функция»;
- научатся создавать линейные алгоритмы на языке Python, содержащие инструкции ввода информации с клавиатуры, вывода информации на экран, математические операции и операции со строками.

Артефакты: консольное приложение для расчета температуры на некоторой высоте по известным данным о температуре на определенной высоте в той же точке. Исходные данные вводятся пользователем с клавиатуры. Результаты выводятся на экран.

Процедура и форма выявления образовательного результата

Представление результатов образовательной деятельности пройдет в форме публичной презентации решений кейса командами и последующих ответов выступающих на вопросы наставника и других команд.

Необходимые материалы и оборудование

Для преподавателя: персональный компьютер, медиапроектор.

Для обучающихся: персональный компьютер на каждого, флеш-накопитель (по желанию), доступ в сеть Интернет.

Расходные материалы

Для обучающихся: лист бумаги, ручка.

Программное обеспечение

- Язык программирования Python 3. (<https://www.python.org/downloads/>)
- интегрированная среда разработки и обучения на языке Python IDLE (устанавливается вместе с языком программирования Python)

Литература:

1. Статья «Питон, пожирающий мир: как побочный проект одного разработчика стал популярнейшим языком программирования»:

<https://habr.com/ru/post/466807/>

2. Официальная документация для языка программирования Python:
<https://docs.python.org/3.9/>

3. Интересные факты о программировании на Python:
<https://xakep.ru/2017/08/22/geekbrains-python-promo/>
<https://techrocks.ru/2020/06/27/15-facts-about-python-programming/>
<https://itproger.com/news/10-interesnih-faktov-pro-python>

Приложения:

Исходный код варианта реализации кейса

```
print ("Начало работы программы")
print()

h1 = float(input("Введите высоту, температура на которой вам известна (в метрах): "))
t1 = float(input("Введите температуру на этой высоте (в градусах Цельсия): "))
h2 = float(input("Введите высоту, температуру на которой нужно найти (в метрах): "))

dt = 6

t2 = t1 + (h1 - h2) / 1000 * dt

print ("Температура на высоте " + str(h2) + " м составит " + str(t2) + " градусов Цельсия.")

print()
print ("Работа программы завершена")
```

Варианты задач для решения с помощью разработанного калькулятора

1. На вершине горы -5°C , высота горы 4500 м. Определить температуру у подножия горы? Ответ: 22°C .

2. Определите температуру воздуха на вершине горы 3 км, если у подножия горы она составила $+12^{\circ}\text{C}$. Ответ: -6°C .

3. Какова температура воздуха на вершине Памире, если в июле у подножия она составляет $+36^{\circ}\text{C}$? Высота Памира около 6 км. Ответ: 0°C .

4. Определите температуру воздуха за бортом самолета, если температура воздуха у поверхности земли равна 31°C , а высота полета – 5 км? Ответ: 1°C .

Кейс № 2

«Let's travel»

Текст кейса

Вы наверняка пользовались сервисами Google Карты или Яндекс.Карты и знаете, что в них есть возможность определить длительность маршрута по нескольким точкам.

В этом кейсе вам предстоит познакомиться с языком программирования Python и составить свою первую программу, которая позволит рассчитывать длительность прогулки по выбранному маршруту в разном темпе (почитать о том, как зависят количество шагов в минуту и длина шага от темпа движения можно [здесь](#)).

На Google Картах или Яндекс.Картах в любом городе (возможно, именно туда вы мечтаете поехать) выберите 2 точки, между которыми вы хотели бы совершить прогулку. В помощью встроенных инструментов сервисов измерьте расстояние между ними по прямой и с учетом дорог.

Выберите темп передвижения (не менее двух вариантов). А определить длительность прогулки в каждом из случаев вам поможет язык программирования Python. Также рассчитайте, сколько бы длилась ваша прогулка в каждом из случаев, если бы вы двигались между точками по прямой.

Описание продукта

Калькулятор длительности прогулки пешком в разном темпе.

Исходные данные вводятся пользователю с клавиатуры. Результаты выводятся на экран.

Для ввода и вывода информации используйте поясняющие надписи. Уведомляйте пользователя о старте и завершении программы.

Дополнительно: калькулятор длительности поездки на автомобиле, затрат топлива и финансов. Величину среднего расхода топлива на 100 км для выбранного автомобиля и цену выбранного вида топлива на АЗС в регионе можно также узнать в сети Интернет.

Категория кейса: приветственный.

Вопросы к кейсу:

1. Что такое программирование?
2. Для чего придумали языки программирования?

3. Кто и когда придумал язык программирования Python и зачем, если до этого уже существовало много языков программирования?
4. Какие фишки есть у языка программирования Python?
5. Где сейчас используется язык программирования Python?

Количество учебных часов/занятий, на которые рассчитан кейс:

7 ак. ч / 3,5 занятия

Учебно-тематическое планирование (занятие — 2 часа):

Занятие 1	
Цель: Познакомиться, презентовать курс, сформировать у обучающихся понимание, что такое программирование и зачем придумали языки программирования. Установить Python, познакомить обучающихся с работой в интерактивном режиме (REPL, от англ. read-eval-print loop — «цикл „чтение — вычисление — вывод“»), показать, какие операции могут быть выполнены над целыми и действительными числами, а также несколько операций над строками.	
Что делает педагог:	Что делают дети:
1. Приветствие. Представление педагога. Игра на знакомство. Предлагает обучающимся сесть в круг. Берет несколько салфеток из пачки, пускает пачку по кругу и предлагает обучающимся сделать то же самое - каждому взять несколько салфеток из пачки. Затем рассказывает столько фактов о себе, сколько салфеток у него в руке, и предлагает обучающимся сделать то же самое.	Садятся в круг, берут несколько салфеток из пачки, рассказывают столько фактов о себе, сколько салфеток в пачке.
2. Представление курса и обмен ожиданиями.	Рассказывают, как они попали на этот курс, и какие у них цели на этот курс.
3. Делит участников на команды по 3-4 человека: по желанию, жеребьевкой	Делятся на команды предложенным педагогом способом.
3. Раскладывает вниз рубашкой заранее подготовленные карточки с вопросами к кейсу: - Что такое программирование? - Для чего придумали языки программирования? - Кто и когда придумал язык программирования Python и зачем, если до этого уже существовало много языков программирования? - Какие фишки есть у языка программирования Python? - Где сейчас используется язык программирования Python? Резюмирует ответы команд участников и добавляет необходимую информацию.	Команды участников вытягивают вопрос, на который им предстоит искать ответ, и ищут на него ответ, используя ресурсы, предложенные педагогом, или другие ресурсы сети Internet (10–15 минут). Затем команды участников презентуют свои ответы.
2. Устанавливает Python в реальном времени или показывает и рассказывает обучающимся, как это сделать: https://www.python.org/downloads/ (для экономии времени можно скачать дистрибутив	Повторяют действия за педагогом на своих персональных компьютерах.

Python заранее). Проверяет или показывает, как проверить, что Python установлен на компьютере (меню «Пуск» → папка «Python 3.9»)	
3. Запускает интерпретатор Python (Python 3.9 (64-bit)), показывает, что он может работать как «умный» калькулятор на ряде примеров. Сложение и вычитание: – Python умеет складывать и вычитать, в том числе в ответе могут получиться отрицательные числа; - если нужно сложить или вычесть отрицательное число, то скобки можно писать или не писать. Предлагает обучающимся придумать свои примеры на сложение и вычитание: - чтобы в конце получилось отрицательное число - чтобы участвовали отрицательные числа. Резюмирует.	Повторяют действия за педагогом на своих персональных компьютерах. Повторяют действия за педагогом на своих персональных компьютерах. Придумывают свои примеры на сложение и вычитание, чтобы в конце получилось отрицательное число. Придумывают свои примеры на сложение и вычитание с отрицательными числами.
4. Показывает, что интерпретатор Python может работать как «умный» калькулятор на ряде примеров (продолжение). Умножение и деление: - Python умеет умножать и делить, причем в ответе могут получиться дробные числа; - в качестве разделителя в языке программирования Python используется точка. Определяет понятия оператора и операнда. Предлагает обучающимся придумать свои примеры такие, чтобы: – в ответе получилось число с дробной частью; – числа с дробной частью использовались в сложении, вычитании, умножении, делении; - проверить, что происходит при делении на ноль. Резюмирует.	Придумывают свои примеры умножение и деление, в том числе такие, чтобы - в ответе получилось число с дробной частью; - числа с дробной частью использовались в сложении, вычитании, умножении, делении.
5. Показывает, что интерпретатор Python может работать как «умный» калькулятор на ряде примеров (продолжение). Скобки. Предлагает обучающимся расставить скобки, чтобы получились верные равенства: $9+9+9/9+9+9+9 = 10$ $99-9:9*9+9:9=100$ Предлагает обучающимся придумать свои примеры, такие, где от расстановки скобок зависит результат (обучающимся нужно вывести несколько вариантов: со скобками и без, с разными способами расстановки скобок).	Пробуют расставить скобки в примерах преподавателя для получения верного равенства. Придумывают свои примеры со скобками такие, где от расстановки скобок зависит результат.
6. Предлагает обучающимся вспомнить про приоритет математических действий и назвать их в порядке убывания приоритета. Фиксирует операторы по убыванию приоритета на доске или флипчарте (можно в виде таблицы со столбцами "Приоритет", "Оператор", "Математическая	Обучающиеся вспоминают математические операции и их приоритет.

операция") () скобки ** возведение в степень +x, -x, ~x унарные плюс, минус *, /, //, % умножение, деление, частное от деления с остатком, взятие остатка +, - сложение и вычитание	
7. Показывает, что интерпретатор Python может работать как «умный» калькулятор на ряде примеров (продолжение). Возведение в степень и извлечения корня. Обращает внимание, что при извлечении корня нужно использовать скобки, так как оператор возведения в степень имеет больший приоритет, чем другие операторы.	Придумывают свои примеры на возведение в степень и извлечение корня, в том числе «неквадратного»
7. Говорит о том, в программах на самых разных языках программирования кроме чисел, часто используются строки. Например, чтобы вывести пользователю приглашение для ввода или прокомментировать вывод. - В языке программирования Python для обозначения строк используются одинарные (апострофы) или двойные кавычки. - Расстояние между словами отражается пробелом. - Знак препинания - тоже часть строки. Строка может состоять только из одного знака препинания или только из нескольких знаков препинания. - Строки могут содержать любые печатные символы, которые мы вводим с клавиатуры (стоит аккуратнее быть с кавычками внутри строк и обратным слешем \ - об этом будет рассказано позже). Предлагает обучающимся сформировать определенные строки: - строку, содержащую их имя (с заглавной буквы) и обратить внимание, сохраняет ли Python регистр; - строку, содержащую их имя, отчество и фамилию с пробелами между ними; - строки, содержащие только русские, только английские, и те и те буквы; - пустую строку; - строку из одного пробела; - строку из нескольких пробелов; - строку из одного знака препинания; - строку из символов не букв и не цифр; - строку, симметричную относительно своего центра (например, "%\$шалаш\$", можно напомнить о существовании таких слов и фраз как палиндромы); - строку с определенным порядком (например, "АббААбббАААбббб"). Предлагает обучающимся применить к строкам	Обучающиеся формируют строки: - строку, содержащую их имя (с заглавной буквы) и обратить внимание, сохраняет ли Python регистр; - строку, содержащую их имя, отчество и фамилию с пробелами между ними; - строки, содержащие только русские, только английские, и те и те буквы; - пустую строку; - строку из одного пробела; - строку из нескольких пробелов; - строку из одного знака препинания; - строку из символов не букв и не цифр; - строку, симметричную относительно своего центра (например, "%\$шалаш\$", можно напомнить о существовании таких слов и фраз как палиндромы); - строку с определенным порядком (например, "АббААбббАААбббб").

операцию сложения (конкатенации) и дублирования ("умножения"): - составить предложение из отдельных слов и знаков препинания; - продублировать строки (например, если продублировать строки "МЫШКА", "БАНКА", "КУПИЛ", "СЫРО" в получившейся строке можно увидеть другие слова - камыш, кабан, пилка, росы). Резюмирует.	
8. Резюмирует, что было сделано на занятии.	Дополняют резюме преподавателя
9. Предлагает обучающимся высказать свои впечатления от занятия.	Высказывают свои впечатления от занятия
Занятие 2	
Цель: Познакомить обучающихся с интерактивной средой разработки IDLE. Объяснить, что такое переменные и типы данных и зачем они нужны. Рассказать о том, зачем и как сохранять программы, написанные на языке программирования Python в файл. Рассказать, как выводить, как запрашивать у пользователя информацию на примере задачи о расчете периметра и площади прямоугольника. Самостоятельная работа над кейсом (продолжение).	
Что делает педагог:	Что делают дети:
1. Напоминает, чему было посвящено предыдущее занятие. Озвучивает цель и план занятия: - продолжение знакомства друг с другом (игра «Посылка»); - продолжение знакомства с языком программирования Python; - знакомство с кейсом, при необходимости разделение на пары / команды; - самостоятельная работа над кейсом, консультации преподавателя; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой.	
2. Проводит игру на продолжение знакомства «Посылки» и помогает всем обучающимся активно включиться в игру. Игра проводится в круге (лучше сидя, но можно и стоя). Для игры понадобится небольшой мячик. Преподаватель называет себя, называет по имени любого участника, кроме тех, кто рядом с ним, придумывает ему посылку на букву, с которой начинается его имя и кидает ему мяч. Далее посылку отправляет уже этот участник. Посылки каждый раз разные. На первую букву имени получателя. Отправлять посылку дважды одному участнику нельзя. Посылки не должны повторяться. Когда последний участник получит посылку, можно сделать возврат посылок по тому же маршруту. Обычно это интересно и проходит при активной	Участвуют в игре. При необходимости помогают другим участникам вспомнить имена.

поддержке участников, так как нужно вспомнить не только того, кто слал тебе посылку, но и что именно он тебе слал.	
2. Знакомит обучающихся с интерактивной средой разработки IDLE: - как запустить; - интерактивный режим.	Выполняют несколько действий в интерактивной среде разработки IDLE. Обсуждают неудобство использования интерактивного режима при программном решении сложных задач.
3. Дает определение встроенных функций языка Python и их параметров, приводит примеры: - функция len() - для определения длины строки; - функция print() - для вывода информации на экран, информация может быть текстовой и числовой (выведенные параметры разделяются пробелами, если нужен другой разделитель - используется именованный параметр sep). Предлагает обучающимся определить количество символов в словах: абракадабра, авто электростеклоподъемники, всплеск (7 букв, 1 слог) и др. Вывести на экран любую поговорку, используя одну строку или несколько строк.	Определяют количество символов в словах: абракадабра, авто электростеклоподъемники, всплеск (7 букв, 1 слог) и др. Выводят на экран любую поговорку, используя одну строку или несколько строк.
4. Подводит обучающихся к мысли о неудобстве интерактивного режима. Предлагает обсудить, будет ли удобен интерактивный режим при программном решении сложных задач. Например, необходимо вывести на экран четверостишие, используя функцию print() для каждой строки. Резюмирует: Будет удобнее, если разместить этот код в файле.	Пробуют вывести на экран четверостишие, используя функцию print() для каждой строки.
5. Демонстрирует, как создать новый файл с кодом программы на Python (с расширением *.py), как сохранить его, а затем заново открыть и запустить.	Повторяют действия за преподавателем.
6. Повторно предлагает обучающимся вывести на экран четверостишие, используя функцию print() для каждой строки, и разместить этот код в файле.	Создают файл *.py для вывода на экран четверостишия, используя функцию print() для каждой строки.
7. Подводит обучающихся к мысли о необходимости переменных. Озвучивает задачу: Автомобиль движется по дороге с определенной скоростью - 25 м/с. Определить, сколько км проедет автомобиль за 1 ч, 2ч, 5 ч, 7 ч, 10 ч, 12 ч и т.д. Предлагает увеличить или уменьшить скорость? Что теперь нужно делать с вычислениями? Везде заменять 25 на другую величину - неудобно. Лучше скорость где-то хранить а затем для	Создают файл *.py для расчета пробега автомобиля.

вычислений просто подставлять. Озвучивает определение переменной и механизм создания и использования переменных в Python. Приводит примеры создания переменных разных типов.	
8. Рассказывает, как можно менять тип и между какими типами это возможно.	Проверяют действие функций преобразования типов int(), float(), str(), bool()
9. Подводит обучающихся к мысли о том, что не всегда удобно задавать значения переменных сразу в программу. В ряде случаев значения должен вводить пользователь. Также в ряде случаев необходимо хранить информацию, которая будет использоваться позднее.	Предлагают свои варианты, как поступить в этом случае.
11. Дает определение переменной, рассказывает о том, что Python определяет тип данных переменной, исходя из значения, которое ей присвоено (на профессиональном языке - является языком с динамической типизацией).	Создают несколько переменных разных типов. Проверяют тип переменных функцией type().
12. Рассказывает, как ввести значение с клавиатуры в переменную - с помощью функции input(). Как сделать приглашение для ввода. Значение какого типа вводится по умолчанию (строка - str), как изменить тип вводимого значения. Предлагает ввести 2 значения, один раз привести их к типу int(), а другой раз - к str() и сложить (результат будет разным).	Пробуют вводить значения в переменные с клавиатуры и изменять тип данных переменных. Вводят с клавиатуры 2 значения, од раз привести их к типу int(), а другой раз - к str() и сложить (результат будет разным).
13. Резюмирует, что умеют делать обучающиеся на языке программирования Python на данный момент.	Дополняют резюме преподавателя.
14. Озвучивает кейс для самостоятельной работы и форму презентации результатов по нему. При необходимости предлагает обучающимся разбиться на пары / команды.	При необходимости разбиваются на пары / команды.
15. Озвучивает необходимость составления плана работ и назначения ответственных (на листе бумаге, на флипчарте с помощью стикеров, в программном обеспечении Trello или другом).	Выбирают средство планирования, составляют план работ и назначают ответственных (при работе в паре / команде)
16. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости.	Самостоятельно работают над кейсом
17. Предлагает обучающимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.
Занятие 3	
Цель: Самостоятельная работа над кейсом (продолжение). Подготовка материалов для презентации результатов кейса	

Что делает педагог:	Что делают дети:
1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; В дальнейшем следит за таймингом.	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут способы преодоления возникших трудностей.
2. Наблюдает за продвижением участников / пар / команд, предупреждает возникновение конфликтов, консультирует по мере необходимости. Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на интересующую тему, или натолкнуть его на мысль, как получить результат на основе тех знаний, которые у него есть в данный момент.	Самостоятельно работают над кейсом.
3. Озвучивает необходимость перейти к подготовке презентации для защиты результатов проекта	Готовят презентацию для защиты результатов кейса.
4. Предлагает обучающимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут способы преодоления возникших трудностей.
Занятие 4 (1 час)	
Цель: Провести демонстрацию полученного за кейс результата, провести рефлекссию	
Что делает педагог:	Что делают дети:
1. Озвучивает цель занятия, определяет порядок демонстрации результатов кейса. Например, на выступление каждой команды не более 3-х минут, далее в течение 2-х минут вопросы.	
2. Устанавливает порядок, в котором обучающиеся будут демонстрировать свои результаты, или порядок может быть случайным: раскладывает на столе карточки с номером выступления	Каждый участник / команда берет карточку с номером выступления
3. Осуществляет модерацию презентаций результатов кейса участниками.	Участники / команда презентуют результаты кейса
4. Проводит рефлексию (лучше проводить, сидя в кругу). Задает вопросы обучающимся:	Отвечают на вопросы преподавателя:

1. Расскажите, чему вы научились за этот кейс, какие навыки освоили. 2. Удалось ли применить полученные навыки еще где-то? (возможно, вы установили интерпретатор Python и интегрированную среду разработки IDLE на своем домашнем компьютере / ноутбуке и показали работу интерпретатора Python своим друзьям / родителям / братьям и сестрам). 3. Что в этом кейсе далось вам легко, а что было трудным для освоения? Как вы справились с этими трудностями?	1. Чему научились за этот кейс, какие навыки освоили. 2. Удалось ли применить полученные навыки еще где-то? . 3. Что в этом кейсе далось легко, а что было трудным для освоения? Как справились с этими трудностями?
--	--

Минимально необходимый уровень входных компетенций: базовое владение ПК.

Предполагаемые результаты обучающихся, формируемые навыки:

- обучающиеся научатся взаимодействовать с интерпретатором Python, создавать и запускать простые программы на языке Python в файлах с расширением *.py в интегрированной среде разработки IDLE;
- освоят понятия «язык программирования», «среда программирования», «скрипт», «инструкция», «операция», «линейный алгоритм», «переменная», «тип данных», «встроенная функция»;
- научатся создавать линейные алгоритмы на языке Python, содержащие инструкции ввода информации с клавиатуры, вывода информации на экран, математические операции и операции со строками.

Артефакты: консольное приложение – калькулятор длительности прогулки пешком в разном темпе. Исходные данные вводятся пользователю с клавиатуры. Результаты выводятся на экран.

Процедура и форма выявления образовательного результата

Представление результатов образовательной деятельности пройдет в форме публичной презентации решений кейса командами и последующих ответов выступающих на вопросы наставника и других команд.

Необходимые материалы и оборудование

Для преподавателя: персональный компьютер, медиапроектор.

Для обучающихся: персональный компьютер на каждого, флеш-накопитель (по желанию), доступ в сеть Интернет.

Расходные материалы

Для обучающихся: лист бумаги, ручка.

Программное обеспечение

- Язык программирования Python 3. (<https://www.python.org/downloads/>)
- интегрированная среда разработки и обучения на языке Python IDLE (устанавливается вместе с языком программирования Python)

Литература:

1. Статья «Питон, пожирающий мир: как побочный проект одного разработчика стал популярнейшим языком программирования»:

<https://habr.com/ru/post/466807/>

2. Официальная документация для языка программирования Python:

<https://docs.python.org/3.9/>

3. Интересные факты о программировании на Python:

<https://xakep.ru/2017/08/22/geekbrains-python-promo/>

<https://techrocks.ru/2020/06/27/15-facts-about-python-programming/>

<https://itproger.com/news/10-interesnih-faktov-pro-python>

Приложения:

Исходный код варианта реализации кейса

```
print ("Начало работы программы")
print()

s1 = int(input("Введите длину маршрута по прямой (в метрах): "))
s2 = int(input("Введите длину маршрута с учетом улиц (в метрах): "))

temp1 = 84.5 # темп замедленный, шагов в минуту
step1 = 0.6 # длина шага (м)

temp2 = 125 # темп ускоренный, шагов в минуту
step2 = 0.84 # длина шага (м)

t1 = s2 // (temp1 * step1)
t2 = s2 // (temp2 * step2)

t3 = s1 // (temp2 * step2)

print()
print ("В замедленном темпе ваша прогулка займет", str(t1), "мин")
print ("В ускоренном темпе ваша прогулка займет", str(t2), "мин")

print ("Если бы Вы шли быстрым шагом по прямой, вам понадобилось бы ", str(t3),
"мин")

print()
print ("Работа программы завершена")
```


Вариант исходных данных

1. Длина маршрута составляет 4,5 км.
2. Расход топлива (бензина АИ-92) для автомобиля Лада Веста на 100 км 7,8 л.
Стоимость 1 л бензина АИ-92 – 42 руб.

Кейс № 3

«Угадай число»

Текст кейса

Вы знаете игру «Угадай число»? Возможно, вы даже играли в нее со своими родителями или друзьями. Один игрок загадывает число. Например, в диапазоне от 1 до 100. А другому игроку нужно его угадать: он предлагает варианты, а загадавший отвечает, было ли его число больше или меньше загаданного. Угадывающий выигрывает, если успевает угадать число за определенное число попыток.

В этом кейсе вместе с преподавателем вам предстоит написать программу, которая позволит вам играть с компьютером. Он будет загадывать числа, а Вы угадывать за определенное число попыток.

А затем вы напишете свой тренажер умножения или сложения. Числа в нем будут генерироваться автоматически, а результат пользователю придется считать в уме, а затем вводить на проверку.

Описание продукта

По примеру преподавателя. Игра «Угадай число» в виде консольного приложения. Пользователь задает диапазон чисел. Компьютер случайным образом выбирает из этого диапазона число и загадывает. Пользователь пытается угадать число за определенное число попыток. Число попыток рассчитывается автоматически в зависимости от диапазона.

Предполагаемые числа вводятся пользователем с клавиатуры. Результаты о том, было ли предполагаемое число больше или меньше загаданного, выводятся на экран после каждой попытки.

В рамках каждой игры можно проводить несколько раундов и определять лучший результат – минимальное число попыток, за которое было угадано число.

Самостоятельная часть кейса. Консольный тренажер для определенного математического действия (умножение, сложение и др.). Количество примеров задается пользователем. Примеры генерируются автоматически в заданном диапазоне. Ответ на пример вводится пользователем с клавиатуры. Результат проверки и правильный ответ (в случае неправильного ответа) выводятся сразу после ввода ответа. Результаты выводятся на экран.

Подсчитываются количество решенных примеров, число правильных и неправильных решений. После завершения тестирования эта информация также выводится пользователю на экран.

Для ввода и вывода информации используйте поясняющие надписи. Уведомляйте пользователя о старте и завершении программы.

Категория кейса: вводный

Вопросы к кейсу:

1. Как в играх ведется подсчет очков? Какие варианты (алгоритмы) возможны?
2. Представьте, что Вы разрабатываете свою игру. Вам нужно хранить информацию о количестве набранных игроком очков. Каким образом бы вы это делали? Какие конструкции языка программирования вам бы понадобились?

Количество учебных часов/занятий, на которые рассчитан кейс: 8 ак. ч / 4 занятия

Учебно-тематическое планирование (занятие — 2 часа):

Занятие 1	
Цель: Познакомить обучающихся с понятием модуля в языке программирования Python, со случайными числами, алгоритмом ветвления и циклическим алгоритмом и их реализацией на языке Python. Совместно с обучающимися реализовать игру "Угадай число", постепенно наращивая ее функционал.	
Что делает педагог:	Что делают дети:
1. Предлагает обучающимся немного поиграть перед началом работы над вторым кейсом. Игра "Половинки" (на развитие навыков коммуникации и выявление лидера). Для игры понадобятся картинки, разрезанные пополам. Количество кусочков должно быть равно числу участников. До начала игры картинки раскладываются на столе обратной стороной вверх. Участники берут по одному кусочку и по сигналу преподавателя ищут пару к картинке. Если группа успевает собрать все картинки за 1 минуту (30 с, 90 с - время устанавливается в зависимости от числа участников), она получает очко. Проводятся несколько раундов, интервал времени каждый раз уменьшается, при этом также можно увеличивать количество очков за раунд. Примечание к игре. Если участников нечетное число, в игре может участвовать и преподаватель или одному из участников можно поручить организационную роль: следить за временем и записывать очки. После того, как участники собрали пару, они могут помогать другим участникам найти пару, что свидетельствует об их достаточно высокой	Участвуют в игре. При дальнейшем выполнении кейса в парах составляют пары, как в последнем раунде.

коммуникативности и ориентации на коллектив. Также в процессе игры некоторые участники могут попытаться организовать процесс, что свидетельствует об их лидерских качествах. После окончания игры фиксирует результат: столько-то очков из максимально возможного. Если выполнение кейса планируется в парах, последние получившиеся пары могут стать рабочими на данный кейс.	
2. Спрашивает или сам напоминает, чему научились обучающиеся во время работы над приветственным кейсом. Знакомит обучающихся со вторым кейсом.	Вспоминают, чему научились во время работы над первым кейсом.
3. Спрашивает обучающихся, знают ли они, что такое случайные числа. Определяет понятие случайного числа (можно на примере игрального кубика, на занятие можно принести настоящий игральный кубик).	Рассказывают то, что знают о случайных числах.
4. Обращает внимание обучающихся на то, в задачах по программированию довольно часто бывает нужно получить случайное число. Причем не только в диапазоне от 1 до 6, как с игральным кубиком. Но в самых разных диапазонах. Например, от 1 до 100 или от 1000 до 2000. Рассказывает о функции - randint(), которая позволяет получить целое число в заданном диапазоне (желательно пояснить, почему у функции такое название). Создает новый файл *.py и пытается вызвать функцию randint(1, 6), чтобы получить случайное значение в диапазоне от 1 до 6, как с игральным кубиком, в переменную с каким-то именем, но получает ошибку.	Повторяют действия за преподавателем: создают новый файл *.py и пытаются вызвать функцию randint(1, 6), чтобы получить случайное значение в диапазоне от 1 до 6, как с игральным кубиком, в переменную с каким-то именем, но получают ошибку, как и преподаватель.
5. Рассказывает учащимся о том, что Python умеет делать очень много. Но не все возможности Python используются в каждой программе. Поэтому ряд возможностей Python вынесены в отдельные модули. Чтобы добавить эти возможности (переменные, функции) в свою программу, используется инструкция import. Показывает, как импортировать модуль целиком и как импортировать только часть модуля. Предлагает учащимся получить случайные числа в разных диапазонах, а также проверить, может ли функция randint() генерировать отрицательное случайное число (подсказка: можно, но нужно правильно записать диапазон - от наименьшего отрицательного числа к большему). Предлагает учащимся генерировать отрицательное случайное число, не используя диапазон с границами	Импортируют модуль random целиком, запускают программу. Импортируют только функцию randint из модуля random и запускают программу. Генерируют случайные числа в разных диапазонах, выводят их на экран. Проверяют, может ли функция randint() генерировать отрицательное случайное число. Ищут способ сгенерировать отрицательное случайное число, не используя диапазон с

- отрицательными числами (подсказка: можно использовать диапазон с границами - положительными числами, а затем добавить знак минус).	границами - отрицательными числами).
6. Предлагает обучающимся диалог с пользователем в начало программы. Когда учащиеся в большинстве своем справятся с этой задачей, реализует этот функционал в своем примере.	Добавляют в начало программы диалог с пользователем.
7. Предлагает обучающимся самостоятельно реализовать ввод пользователем своей догадки. Затем реализует этот функционал в своем примере.	Добавляют в программу ввод данных пользователем.
8. Спрашивает, каким образом будет происходить проверка? После ответов обучающихся резюмирует.	Предлагают свои варианты.
9. Рисует на доске или флипчарте блок-схему проверки ввода пользователя. Определяет алгоритм ветвления. Показывает, как записать алгоритм ветвления для проверки "угадал - не угадал", используя несколько условных операторов if. Показывает, как записать алгоритм ветвления для проверки "угадал - не угадал", используя условный оператор if-else. Напоминает условие задачи: после попытки ответа пользователя нужно, сообщать пользователю, больше или меньше заданное число. Показывает, как записать алгоритм ветвления для проверки "угадал - больше - меньше", используя несколько условных операторов if. Показывает, как записать алгоритм ветвления для проверки "угадал - больше - меньше", используя условный оператор if-elif-else.	Реализуют все способы вслед за преподавателем, комментируют или сохраняют несколько файлов с разными вариантами.
10. Предлагает дать пользователю несколько попыток угадать число, то есть несколько раз придется сделать одно и то же. Писать несколько раз одни и те же условные операторы - не самый лучший вариант. Если действия повторяются, необходимо использовать циклы. Определяет циклический алгоритм. Рисует на доске или флипчарте блок-схему циклического алгоритма. Показывает, как записать циклический алгоритм, используя цикл for (в функции range()) пока останавливаться не стоит, необходимо, чтобы написанный код работал). Рассказывает об операторе break для преждевременного выхода из цикла в случае, если пользователь угадал число.	Добавляют новый код в программу вслед за преподавателем. Запускают и отлаживают программу.

11. Рассказывает о встроенной функции range() и ее особенностях. Предлагает обучающимся изменить число попыток.	Экспериментируют с параметрами функции range()
12. Напоминает обучающимся о том, что сообщение об успешной догадке внутри цикла было заменено на оператор break. Предлагает учащимся подумать над тем, как сообщить пользователю об успешной догадке после выхода из цикла (в этот же момент можно вывести информацию о том, с какой попытки было угадано число) или о том, что попытки закончились (в этот же момент можно вывести пользователю информацию о том, какое число было загадано). Резюмирует и реализует этот функционал в своем примере.	Участвуют в обсуждении, реализуют свой вариант проверки. При необходимости повторяют вариант преподавателя.
13. Если остается время, можно показать, как добавить дополнительный функционал: - прекращение работы программы при вводе определенного символа (например, 'q'); - игра несколько раз (на старте можно спросить у пользователя, сколько раз он хочет сыграть); - игра бесконечное число раз (цикл while true); - подсчет количества всех сыгранных игр в случае игры о остановки по требованию и общего числа игр. Этот функционал также можно реализовать в начале второго занятия.	Повторяют за преподавателем или самостоятельно реализуют функционал: - прекращение работы программы при вводе определенного символа (например, 'q'); - игра несколько раз на старте можно спросить у пользователя, сколько раз он хочет сыграть); - игра бесконечное число раз (цикл while true); - подсчет количества всех сыгранных игр в случае игры о остановки по требованию и общего числа игр.
13. Резюмирует, что было сделано на занятии.	Дополняют резюме преподавателя
14. Предлагает учащимся высказать свои впечатления от занятия.	Высказывают свои впечатления от занятия
Занятие 2	
Цель: Познакомить учащихся с самостоятельным заданием на этот кейс. Планирование обучающимися работ над кейсом. Самостоятельная работа обучающихся над кейсом.	
Что делает педагог:	Что делают дети:
1. Озвучивает кейс для самостоятельной работы и требования к нему. Предлагает найти онлайн-тренажеры для сложения, умножения и других математических операций.	Ищут и тестируют онлайн-тренажеры для сложения, умножения и других математических операций.
2. Организует распределение тем тренажеров (например, команды могут вытягивать листочки с вариантом тренажера): - тренажер таблицы умножения;	Получают тему тренажера.

- тренажер квадратов чисел (2*2, 3*3, ..., 10*10, 11*11, ..., 20*20, 25*25 и т.д.); - тренажер умножения числа от 10 до 100 на число от 1 до 10; - тренажер сложения чисел от 1 до 10; - тренажер сложения чисел от 10 до 20; - тренажер сложения чисел от 1 до 100; - аналогично для вычитания, - тренажер деления (в этом случае случайным образом необходимо генерировать делитель и частное, а затем вычислять уменьшаемое) и т.д.	
3. Озвучивает необходимость составления плана работ и назначения ответственных (на листе бумаге, на флипчарте с помощью стикеров, в программном обеспечении Trello или другом). Предлагает сформулировать, что будет входными данными, и что будет на выходе программы. Наблюдает за работой обучающихся, при необходимости консультирует.	Выбирают средство планирования, составляют план работ и назначают ответственных (при работе в паре).
4. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости.	Самостоятельно работают над кейсом.
5. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 3	
Цель: Самостоятельная работа над кейсом (продолжение). Подготовка материалов для презентации результатов кейса	
Что делает педагог:	Что делают дети:
1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой. В дальнейшем следит за таймингом.	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.
2. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости. <i>Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на интересующую тему, или натолкнуть его на мысль, как получить результат</i>	Самостоятельно работают над кейсом

<i>на основе тех знаний, которые у него есть в данный момент.</i>	
3. Озвучивает необходимость перейти к подготовке презентации для защиты результатов проекта	Готовят презентацию для защиты результатов кейса
4. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 4	
Цель: Провести демонстрацию полученного за кейс результата, провести рефлексию	
Что делает педагог:	Что делают дети:
1. Озвучивает цель занятия, определяет порядок демонстрации результатов кейса. Например, на выступление каждой команды не более 5-и минут, далее в течение 3-х минут вопросы.	
2. Устанавливает порядок, в котором учащиеся будут демонстрировать свои результаты. Например, по убыванию суммы букв в имени (и фамилии) участников команд.	Команды считают сумму букв в имени (и фамилии) участников и определяют порядок выступления.
3. Осуществляет модерацию презентаций результатов кейса участниками.	Участники / команда презентуют результаты кейса
4. Проводит рефлексию (лучше проводить, сидя в кругу). Задаёт вопросы учащимся: 1. Расскажите, чему вы научились за этот кейс, какие навыки освоили. 2. Удалось ли применить полученные навыки еще где-то? . 3. Что в этом кейсе далось вам легко, а что было трудным для освоения? Как вы справились с этими трудностями?	Отвечают на вопросы преподавателя: 1. Чему научились за этот кейс, какие навыки освоили. 2. Удалось ли применить полученные навыки еще где-то? . 3. Что в этом кейсе далось легко, а что было трудным для освоения? Как справились с этими трудностями?

Минимально необходимый уровень входных компетенций:

- запуск интегрированной среды разработки IDLE, создание и запуск программы на языке Python в файлах с расширением *.py;
- ввод с клавиатуры, вывод на экран на языке программирования Python;
- создание переменных, присваивание переменным значений;
- математические операции и операции со строками на языке программирования Python.

Предполагаемые результаты обучающихся, формируемые навыки:

- закрепят ввод информации с клавиатуры в переменные, вывод информации на экран, математические операции и операции со строками на языке программирования Python;
- освоят понятия «логический тип данных», «условный оператор», «цикл», «преждевременное завершение программы», «разветвляющийся алгоритм», «циклический алгоритм»;
- познакомятся с операциями сравнения в Python;
- научатся создавать разветвляющиеся алгоритмы на языке Python, простейшие циклические алгоритмы.

Артефакты:

1. консольная игра «Угадай число». Исходные данные вводятся пользователем с клавиатуры. Результаты выводятся на экран.
2. консольный тренажер для определенного математического действия (умножение, сложение и др.). Примеры генерируются автоматически. Настройки и ответ например вводятся пользователем с клавиатуры. Результаты выводятся на экран.

Процедура и форма выявления образовательного результата

Представление результатов образовательной деятельности пройдет в форме публичной презентации решений кейса командами и последующих ответов выступающих на вопросы наставника и других команд.

Необходимые материалы и оборудование

Для преподавателя: персональный компьютер, медиапроектор.

Для обучающихся: персональный компьютер на каждого, флеш-накопитель (по желанию), доступ в сеть Интернет.

Расходные материалы

Для обучающихся: лист бумаги, ручка.

Программное обеспечение

- Язык программирования Python 3. (<https://www.python.org/downloads/>)
- интегрированная среда разработки и обучения на языке Python IDLE (устанавливается вместе с языком программирования Python)

Литература:

1. Условный оператор языка программирования Python:

<https://pythonworld.ru/osnovy/instrukciya-if-elif-else-proverka-istinnosti-trekmestnoe-vyrazhenie-ifelse.html>

2. Цикл for, функция range(), оператор break в языке программирования Python:

<https://pythonworld.ru/osnovy/cikly-for-i-while-operator-break-i-continue-volshebnoe-slovo-else.html>

Приложения:

Исходный код игры «Угадай число»

```
import random

print('Это игра по угадыванию чисел.')
print()

print('Задай диапазон для чисел.')
a = int(input('Нижняя граница диапазона: '))
b = int(input('Верхняя граница диапазона: '))

print('Сколько попыток тебе нужно, чтобы отгадать число?')
max = int(input())

print()
print('Я буду загадывать числа несколько раз в диапазоне [', a, '; ', b, '].', sep = ")
print('Для отгадывания числа у тебя будет ', max, ' попыток.')
print('Если захочешь прекратить игру - нажми q. Начинаем!')
print()

min= 0

while (True):
    number = random.randint(a, b)
    k = 0
    for k in range(max):
        print('Я загадал число. Попробуй угадать.')
        guess = input()
        if guess == 'q':
            print('Ты прервал игру на самом интересном месте. Я буду скучать. До встречи!')
            print('Твой лучший результат: ' + str(min))
            exit()

        guess = int(guess)
        if guess < number:
            print('Увы, недолет...')
        if guess > number:
            print('Увы, перелет...')
        if guess == number:
            break

    if guess == number:
        k = k+1
```

```
if (min == 0) or (k < min):
    min = k
print('Число угадано! Это', str(number), ' Тебе понадобилось попыток:' + str(k))
print('Твой лучший результат: ' + str(min))
print('Продолжим!')
print()

if guess != number:
    print('Увы. Попытки закончились. Ты так и не сумел разгадать число. Я загадал
число: ' + str(number))
    print('Твой лучший результат: ' + str(min))
    print('Продолжим!')
    print()
```

Кейс № 4

«Навстречу приключениям»

Текст кейса

Вы любите квесты? Бродить по старому заброшенному дому... улочкам города. Это может быть очень увлекательным...

Для того чтобы сделать простую игру-квест, не нужно больших познаний в программировании. Достаточно знать, как работать со случайными числами и как реализовать основные алгоритмы на выбранном языке программирования, остальное подскажет фантазия.

В этом кейсе на примере простого квеста с двумя уровнями и несколькими дверьми вы узнаете, как реализовать основные алгоритмы на языке программирования Python, затем создадите свой текстовый квест, а ваши коллеги его протестируют.

Описание продукта

Игра «Текстовый квест» в виде консольного приложения.

Для ввода и вывода информации используйте поясняющие надписи. Уведомляйте пользователя о старте и завершении программы.

Категория кейса: вводный

Вопросы к кейсу:

Как в играх ведется подсчет очков? Какие варианты (алгоритмы) возможны?

3. Представьте, что Вы разрабатываете свою игру. Вам нужно хранить информацию о количестве набранных игроком очков. Каким образом бы вы это делали? Какие конструкции языка программирования вам бы понадобились?

Количество учебных часов/занятий, на которые рассчитан кейс: 8 ак. ч / 4 занятия

Учебно-тематическое планирование (занятие — 2 часа):

Занятие 1	
Цель: Познакомить обучающихся с понятием модуля в языке программирования Python, со случайными числами, алгоритмом ветвления и циклическим алгоритмом и их реализацией на языке Python. Совместно с обучающимися реализовать текстовый квест, постепенно наращивая его возможности.	
Что делает педагог:	Что делают дети:
1. Предлагает обучающимся немного поиграть перед началом работы над вторым кейсом. Игра "Половинки" (на развитие навыков	Участвуют в игре. При дальнейшем выполнении

коммуникации и выявление лидера). Для игры понадобятся картинки, разрезанные пополам. Количество кусочков должно быть равно числу участников. До начала игры карточки раскладываются на столе обратной стороной вверх. Участники берут по одному кусочку и по сигналу преподавателя ищут пару к картинке. Если группа успевает собрать все картинки за 1 минуту (30 с, 90 с - время устанавливается в зависимости от числа участников), она получает очко. Проводятся несколько раундов, интервал времени каждый раз уменьшается, при этом также можно увеличивать количество очков за раунд. Примечание к игре. Если участников нечетное число, в игре может участвовать и преподаватель или одному из участников можно поручить организационную роль: следить за временем и записывать очки. После того, как участники собрали пару, они могут помогать другим участникам найти пару, что свидетельствует об их достаточно высокой коммуникативности и ориентации на коллектив. Также в процессе игры некоторые участники могут пытаться организовать процесс, что свидетельствует об их лидерских качествах. После окончания игры фиксирует результат: столько-то очков из максимально возможного. Если выполнение кейса планируется в парах, последние получившиеся пары могут стать рабочими на данный кейс.	кейса в парах составляют пары, как в последнем раунде.
2. Спрашивает или сам напоминает, чему научились обучающиеся во время работы над приветственным кейсом. Знакомит обучающихся со вторым кейсом.	Вспоминают, чему научились во время работы над первым кейсом.
3. Спрашивает учащихся, знают ли они, что такое случайные числа. Определяет понятие случайного числа (можно на примере игрального кубика, на занятие можно принести настоящий игровой кубик).	Рассказывают то, что знают о случайных числах.
4. Обращает внимание обучающихся на то, в задачах по программированию довольно часто бывает нужно получить случайное число. Причем не только в диапазоне от 1 до 6, как с игральным кубиком. Но в самых разных диапазонах. Например, от 1 до 100 или от 1000 до 2000. Рассказывает о функции - randint(), которая позволяет получить целое число в заданном диапазоне (желательно пояснить, почему у функции такое название). Создает новый файл *.py и пытается вызвать функцию randint(1, 6), чтобы получить случайное	Повторяют действия за преподавателем: создают новый файл *.py и пытаются вызвать функцию randint(1, 6), чтобы получить случайное значение в диапазоне от 1 до 6, как с игральным кубиком, в переменную с каким-то именем, но получают ошибку, как и преподаватель.

значение в диапазоне от 1 до 6, как с игральным кубиком, в переменную с каким-то именем, но получает ошибку.	
5. Рассказывает учащимся о том, что Python умеет делать очень много. Но не все возможности Python используются в каждой программе. Поэтому ряд возможностей Python вынесены в отдельные модули. Чтобы добавить эти возможности (переменные, функции) в свою программу, используется инструкция import. Показывает, как импортировать модуль целиком и как импортировать только часть модуля. Предлагает учащимся получить случайные числа в разных диапазонах, а также проверить, может ли функция randint() генерировать отрицательное случайное число (подсказка: можно, но нужно правильно записать диапазон - от наименьшего отрицательного числа к большему). Предлагает учащимся генерировать отрицательное случайное число, не используя диапазон с границами - отрицательными числами (подсказка: можно использовать диапазон с границами - положительными числами, а затем добавить знак минус).	Импортируют модуль random целиком, запускают программу. Импортируют только функцию randint из модуля random и запускают программу. Генерируют случайные числа в разных диапазонах, выводят их на экран. Проверяют, может ли функция randint() генерировать отрицательное случайное число. Ищут способ сгенерировать отрицательное случайное число, не используя диапазон с границами - отрицательными числами).
6. Напоминает условие задачи и предлагает учащимся создать две переменные: номер плохой пещеры на первом уровне и номер хорошей пещеры на втором уровне - и получить значения случайным образом (обращает внимание учащихся на правильность задания параметров для функции randint в первом и во втором случае). Когда учащиеся в большинстве своем справятся с этой задачей, реализует этот функционал в своем примере.	Создают две переменные: номер плохой пещеры на первом уровне и номер хорошей пещеры на втором уровне - и получают значения случайным образом.
7. Предлагает обучающимся диалог с пользователем в начало программы. Когда учащиеся в большинстве своем справятся с этой задачей, реализует этот функционал в своем примере.	Добавляют в начало программы диалог с пользователем.
8. Предлагает обучающимся самостоятельно реализовать ввод пользователем своего выбора на первом уровне. Затем реализует этот функционал в своем примере.	Добавляют в программу ввод данных пользователем.
9. Спрашивает, каким образом будет происходить проверка, удачная выбрана пещера или неудачная. После ответов обучающихся резюмирует.	Предлагают свои варианты.
10. Рисует на доске или флипчарте блок-схему проверки ввода пользователя.	Реализуют все способы вслед за преподавателем,

Определяет алгоритм ветвления. Показывает, как записать алгоритм ветвления для проверки "удачная пещера - неудачная пещера", используя несколько условных операторов if. Показывает, как записать алгоритм ветвления для проверки "удачная пещера - неудачная пещера", используя условный оператор if-else.	комментируют или сохраняют несколько файлов с разными вариантами.
11. Предлагает учащимся самостоятельно добавить сообщения об удачном и неудачном выборе пользователя. Затем реализует этот функционал в своем примере.	Добавляют новый код в программу вслед за преподавателем. Запускают и отлаживают программу.
12. Предлагает учащимся самостоятельно реализовать второй уровень для той ветки условного оператора, когда первый уровень успешно пройден. Затем реализует этот функционал в своем примере.	Добавляют новый код в программу вслед за преподавателем. Запускают и отлаживают программу.
13. Предлагает учащимся самостоятельно определить приз для той ветки условного оператора, когда второй уровень успешно пройден. Затем реализует этот функционал в своем примере.	Добавляют новый код в программу вслед за преподавателем. Запускают и отлаживают программу.
14. Предлагает учащимся подумать, как образом можно изменить программу, чтобы запустить ее один раз, а затем играть несколько раз или пока не надоест. Резюмирует. Например, возможны такие варианты: - количество попыток задается в программе; - пользователь играет пока не надоест, для отказа вводит специальный символ; - спрашивать пользователя, хочет ли он сыграть еще раз.	Участвуют в обсуждении.
15. Определяет циклический алгоритм. Рисует на доске или флипчарте блок-схему циклического алгоритма (цикл со счетчиком, цикл с условием). Показывает, как записать циклический алгоритм, используя цикл for и функцию range() и цикл while. Рассказывает об операторе break для преждевременного выхода из цикла в случае, если пользователь ввел специальный символ.	Реализуют все способы вслед за преподавателем, комментируют или сохраняют несколько файлов с разными вариантами. Запускают и отлаживают программу.
16. Предлагает учащимся подумать, как добавить подсчет общего числа очков. Резюмирует. Реализует функционал в своем примере.	Участвуют в обсуждении, реализуют функционал самостоятельно или вслед за преподавателем. Запускают и отлаживают программу.
17. Резюмирует, что было сделано на занятии.	Дополняют резюме преподавателя
18. Предлагает учащимся высказать свои впечатления от занятия.	Высказывают свои впечатления от занятия
Занятие 2	

Цель: Познакомить учащихся с самостоятельным заданием на этот кейс. Планирование обучающимися работ над кейсом. Самостоятельная работа обучающихся над кейсом.	
Что делает педагог:	Что делают дети:
1. Озвучивает кейс для самостоятельной работы и требования к нему.	Ищут и тестируют онлайн-тренажеры для сложения, умножения и других математических операций.
2. Модерирует процесс генерации идей для квеста.	Придумывают идеи для квеста и сценарий.
3. Озвучивает необходимость составления плана работ и назначения ответственных (на листе бумаге, на флипчарте с помощью стикеров, в программном обеспечении Trello или другом). Предлагает сформулировать, что будет входными данными, и что будет на выходе программы. Наблюдает за работой обучающихся, при необходимости консультирует.	Выбирают средство планирования, составляют план работ и назначают ответственных (при работе в паре).
4. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости.	Самостоятельно работают над кейсом.
5. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 3	
Цель: Самостоятельная работа над кейсом (продолжение). Подготовка материалов для презентации результатов кейса	
Что делает педагог:	Что делают дети:
1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой. В дальнейшем следит за таймингом.	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.
2. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости. <i>Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на интересующую тему, или натолкнуть его на мысль, как получить результат</i>	Самостоятельно работают над кейсом.

на основе тех знаний, которые у него есть в данный момент.	
3. Озвучивает необходимость перейти к подготовке презентации для защиты результатов проекта	Готовят презентацию для защиты результатов кейса.
4. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 4	
Цель: Провести демонстрацию полученного за кейс результата, провести рефлекссию	
Что делает педагог:	Что делают дети:
1. Озвучивает цель занятия, определяет порядок демонстрации результатов кейса. Например, на выступление каждой команды не более 5-и минут, далее в течение 3-х минут вопросы.	
2. Устанавливает порядок, в котором учащиеся будут демонстрировать свои результаты. Например, по убыванию суммы букв в имени (и фамилии) участников команд.	Команды считают сумму букв в имени (и фамилии) участников и определяют порядок выступления.
3. Осуществляет модерацию презентаций результатов кейса участниками.	Участники / команда презентуют результаты кейса
4. Проводит рефлексию (лучше проводить, сидя в кругу). Задает вопросы учащимся: 1. Расскажите, чему вы научились за этот кейс, какие навыки освоили. 2. Удалось ли применить полученные навыки еще где-то? 3. Что в этом кейсе далось вам легко, а что было трудным для освоения? Как вы справились с этими трудностями?	Отвечают на вопросы преподавателя: 1. Чему научились за этот кейс, какие навыки освоили. 2. Удалось ли применить полученные навыки еще где-то? . 3. Что в этом кейсе далось легко, а что было трудным для освоения? Как справились с этими трудностями?

Минимально необходимый уровень входных компетенций:

- запуск интегрированной среды разработки IDLE, создание и запуск программы на языке Python в файлах с расширением *.py;
- ввод с клавиатуры, ввод на экран на языке программирования Python;
- создание переменных, присваивание переменным значений;
- математические операции и операции со строками на языке программирования Python.

Предполагаемые результаты обучающихся, формируемые навыки:

- закрепят ввод информации с клавиатуры в переменные, вывод информации на экран, математические операции и операции со строками на языке программирования Python;
- освоят понятия «логический тип данных», «условный оператор», «цикл», «преждевременное завершение программы», «разветвляющийся алгоритм», «циклический алгоритм»;
- познакомятся с операциями сравнения в Python;
- научатся создавать разветвляющиеся алгоритмы на языке Python, простейшие циклические алгоритмы.

Артефакты:

3. консольная игра «Угадай число». Исходные данные вводятся пользователем с клавиатуры. Результаты выводятся на экран.
4. консольный тренажер для определенного математического действия (умножение, сложение и др.). Примеры генерируются автоматически. Настройки и ответ например вводятся пользователем с клавиатуры. Результаты выводятся на экран.

Процедура и форма выявления образовательного результата

Представление результатов образовательной деятельности пройдет в форме публичной презентации решений кейса командами и последующих ответов выступающих на вопросы наставника и других команд.

Необходимые материалы и оборудование

Для преподавателя: персональный компьютер, медиапроектор.

Для обучающихся: персональный компьютер на каждого, флеш-накопитель (по желанию), доступ в сеть Интернет.

Расходные материалы

Для обучающихся: лист бумаги, ручка.

Программное обеспечение

- Язык программирования Python 3. (<https://www.python.org/downloads/>)
- интегрированная среда разработки и обучения на языке Python IDLE (устанавливается вместе с языком программирования Python)

Литература:

3. Условный оператор языка программирования Python:

<https://pythonworld.ru/osnovy/instrukciya-if-elif-else-proverka-istinnosti-trekmestnoe-vyrazhenie-ifelse.html>

4. Цикл for, функция range(), оператор break в языке программирования Python:

<https://pythonworld.ru/osnovy/cikly-for-i-while-operator-break-i-continue-volshebnoe-slovo-else.html>

Приложения:

Исходный код игры «Навстречу приключениям»

```
import random
import time

print('Добро пожаловать в Квест.')
print()

total = 0

playAgain = 'да'
while playAgain == 'да' or playAgain == 'д':
    badCave1 = random.randint(1, 3)
    goodCave2 = random.randint(1, 2)

    print('Ты очутился в подземелье... В нем два уровня. ')
    print('На первом уровне 3 пещеры. И только две из них ведут дальше. В одной - обрыв... Какую пещеру выберешь?')

    cave1 = 0
    while cave1 != 1 and cave1 != 2 and cave1 != 3:
        print('В какую пещеру ты войдешь? (нажми клавишу 1, 2 или 3)')
        cave1 = int(input())

    print('Ты движешься по пещере...')
    time.sleep(2)
    print('Ее темнота заставляет тебя дрожать от страха...')
    time.sleep(2)

    print('Ты спотыкаешься...')
    time.sleep(2)

    if cave1 == badCave1:
        print('... и летишь в пропасть... Эта игра для тебя закончилась... Сожалеем...')
    else:
        print('... и оказываешься в другой пещере. Первый уровень пройден, ты на втором! Поздравляем!')
        time.sleep(2)

    print()
    print('На втором уровне только 2 пещеры. В одной из них - злой дракон, от которого тебе не спастись... В другой - дружелюбный милашка-дракончик, который одарит тебя сокровищами.')
    cave2 = 0
```

```
while cave2 != 1 and cave2 != 2:
    print('В какую пещеру ты войдешь на этот раз? (нажми клавишу 1 или 2)')
    cave2 = int(input())

print('Ты опять движешься по пещере...')
time.sleep(2)
print('Ее темнота заставляет тебя дрожать от страха...')
time.sleep(2)

print('Вдруг почва уходит у тебя из-под ног... Тебя схватил дракон...')
time.sleep(2)

print(goodCave2)

if cave2 == goodCave2:
    print('Тебе несказанно повезло сегодня! Ты попал в пещеру к дружелюбному дракону!')
    prize = random.randint(10, 100)
    total = total + prize
    print('Забирай сокровища:', str(prize), 'монет!')
    print('Твой общий счет: ', str(total), 'монет!')
    time.sleep(2)
else:
    print('Увы... Тебе не повезло... В этой пещере обитает злой дракон... Эта игра для тебя закончилась... Сожалеем...')

print()
print('Попытаешь удачу еще раз? (Введи да или нет)')
playAgain = input()

print('Квест завершен.')
print()
```

Кейс № 5

«Черепашья графика»

Текст кейса

Во всех программах, которые вы делали до этого, информация была текстовой – буквы, цифры и другие символы. А почему бы не порисовать? Но как добавить языку программирования Python такие возможности?

Помните, как вы добавляли в ваши программы возможности генерировать случайные числа, работать с таймером, выполнять сложные математические действия? Модули – вот в чем секрет.

Для рисования тоже есть модуль! Название у него немного смешное – turtle (переводится с английского как «черепашка»).

Именно «черепашка» позволит вам нарисовать на холсте – экране картину по мотивам одного из произведений известных художников. Картину, которая будет вас вдохновлять, вы можете выбрать самостоятельно. Жанр картины – образца может быть любым: кубизм или абстракционизм (с этими жанрами вам будет легче работать), а может быть это будет футуризм или модерн.

Попробуйте продумать процесс рисования картины так, чтобы перед началом пользователь мог сделать «заказ». Так пользователь вашего приложения будет чувствовать себя немного творцом картины. Например, вы можете предложить пользователю указать количество звезд, которое будет на картине, или предложить выбор: использовать только треугольники, или только квадраты.

Если вас увлечет «черепашья графика» и вы успеете написать код для рисования нескольких картин, предложите пользователю выбор, какую картину нарисовать.

Описание продукта

Приложение для рисования, которое последовательно будет создавать картину по мотивам одного из произведений известных художников. Настройки по желанию вводятся с клавиатуры. Вывод осуществляется на экран.

Категория кейса: базовый.

Вопросы к кейсу:

1. Вы уже не раз сталкивались с встроенными функциями языка программирования Python. Можете их перечислить?

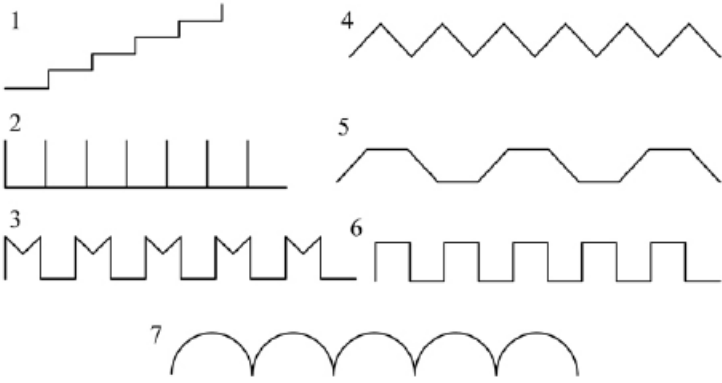
2. Функции часто содержатся в модулях. Назовите несколько функций из модуля `math`, которые выполняют понятные вам действия.
3. Какие у функций особенности? Обязательно ли должны быть параметры у функции? Приведите примеры встроенных функций без параметров и функций, которые выполняют действия, но не возвращают результат.
4. А можно ли написать свою функцию на языке Python, а затем ее использовать? Каким образом это можно сделать? Как интерпретатор поймет, что вы собираетесь описывать функцию, где ее начало и конец. Как передать в функцию параметр, а затем вернуть результат?

Количество учебных часов/занятий, на которые рассчитан кейс: 12 ак. ч / 6 занятий

Учебно-тематическое планирование (занятие — 2 часа):

Занятие 1	
Цель: Познакомить обучающихся с возможностями модуля <code>turtle</code> . Уточнить понятия «функция», «параметры функции», «возвращаемое значение». Научить создавать свои функции на языке программирования Python.	
Что делает педагог:	Что делают дети:
1. Предлагает обучающимся немного поиграть перед началом работы над третьим кейсом. Игра "Что ты знаешь о черепахах" (на развитие навыков коммуникации). Участники делятся на 2 или 3 команды. В течение 5 минут, команды обсуждают, что они знают о черепахах. Это могут быть сведения о живых черепахах, памятниках, мультипликационных героях и т.д. (Можно фиксировать сведения на листе. Пользование интернетом может быть разрешено или запрещено.) Далее команды по очереди называют по одному факту о черепахах. Спикер от команды – не один, каждый раз от команды выступает новый участник. Выигрывает та команд, которая назовет факт последней.	Участвуют в игре.
2. Спрашивает или сам напоминает, чему научились обучающиеся во время работы над приветственным и вводным кейсами. Знакомит обучающихся с третьим кейсом.	Вспоминают, чему научились во время работы над первыми двумя кейсами.
3. Спрашивает, что нужно для рисования. Резюмирует: холст, кисть (или перо, если рисуем тушью).	Обучающиеся высказывают свои варианты.
4. Создает новый файл *.py. Подключает модуль turtle. Создает холст. Задает настройки для холста и исполнителя.	Создает новый файл *.py. Добавляют код в программу вслед за педагогом.
5. Знакомит обучающихся с основными командами для рисования пером (команды приведены в приложении к кейсу).	Добавляют код в программу вслед за педагогом.

6. Показывает, как нарисовать замкнутый контур и как его заполнить (команды приведены в приложении к кейсу) на примере квадрата и треугольника (пока без циклов). Перед тем как писать код желательно показать процесс рисования на флипчарте.	Добавляют код в программу вслед за педагогом.
7. Предлагает немного отвлечься и ответить на вопросы кейса. После ответов учащихся определяет понятия функция», «параметры функции», «возвращаемое значение».	Обучающиеся ищут ответы на вопросы кейса, используя ресурсы сети Интернет. Обсуждают.
8. Для повышения читабельности кода предлагает вынести рисование отдельных фигур в функции. После того как обучающиеся самостоятельно попробуют это сделать, реализует этот функционал.	Создают функции самостоятельно или воспроизводят код педагога.
9. Предлагает учащимся самостоятельно реализовать функции для рисования прямоугольника, ромба, трапеции. Углы можно определять транспортиром по рисунку или математически, если обучающиеся владеют соответствующим математическим аппаратом.	Сначала рисуют фигуру на листе бумаги, определяют углы поворота. Затем создают функции для рисования фигур. Запускают программу и отлаживают.
10. Предлагает учащимся собрать созданные функции в отдельный модуль (файл) и подключить его к своей программе.	Выносят все созданные функции в отдельный файл и подключают его к своей программе. Запускают программу и отлаживают.
17. Резюмирует, что было сделано на занятии.	Дополняют резюме преподавателя
18. Предлагает учащимся высказать свои впечатления от занятия.	Высказывают свои впечатления от занятия
Занятие 2	
Цель: Показать, как циклы могут использоваться для рисования. Познакомить учащихся с самостоятельным заданием на этот кейс. Планирование обучающимися работ над кейсом. Самостоятельная работа обучающихся над кейсом.	
Что делает педагог:	Что делают дети:
1. Предлагает учащимся вспомнить, что было изучено на предыдущем занятии.	Вспоминают, что было изучено на предыдущем занятии.
2. Предлагает обучающимся использовать циклы для рисования простых последовательностей:	Создают новый файл *.ру. Создают программу по заданию: сначала без циклов, затем заменяют повторяющиеся строки циклами.

	
3. Предлагает обучающимся использовать циклы для рисования правильных многоугольников: равностороннего треугольника, квадрата, пятиугольника, шестиугольника. Рисование каждой фигуры желательно вынести в отдельную функцию. После того как обучающиеся попробуют сделать это сами, предлагает свое решение.	Создают новый файл *.ру. Создают функции по заданию. Запускают и отлаживают. При необходимости обращаются к коду педагога.
4. Предлагает учащимся найти закономерности в рисовании правильных многоугольников. После обсуждения резюмирует. Реализует код функции для рисования правильного n-угольника.	Ищут закономерности в рисовании правильных многоугольников, обсуждают. Создают функцию для рисования правильного n-угольника.
5. Озвучивает кейс для самостоятельной работы и требования к нему.	Ищут и тестируют онлайн-тренажеры для сложения, умножения и других математических операций.
6. Модерирует процесс генерации идей для будущей картины.	Придумывают идею для будущей картины.
7. Озвучивает необходимость составления плана работ и назначения ответственных (на листе бумаге, на флипчарте с помощью стикеров, в программном обеспечении Trello или другом). Предлагает сформулировать, что будет входными данными, и что будет на выходе программы. Наблюдает за работой обучающихся, при необходимости консультирует.	Выбирают средство планирования, составляют план работ и назначают ответственных (при работе в паре).
8. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости.	Самостоятельно работают над кейсом.
9. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 3	
Цель: Самостоятельная работа над кейсом (продолжение).	
Что делает педагог:	Что делают дети:

1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой. В дальнейшем следит за таймингом.	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.
2. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости. <i>Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на интересующую тему, или натолкнуть его на мысль, как получить результат на основе тех знаний, которые у него есть в данный момент.</i>	Самостоятельно работают над кейсом
3. Озвучивает необходимость перейти к подготовке презентации для защиты результатов проекта	Готовят презентацию для защиты результатов кейса
4. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 4	
Цель: Самостоятельная работа над кейсом (продолжение)	
Что делает педагог:	Что делают дети:
1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой. В дальнейшем следит за таймингом.	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.
2. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости. <i>Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на интересующую тему, или натолкнуть его на мысль, как</i>	Самостоятельно работают над кейсом

получить результат на основе тех знаний, которые у него есть в данный момент.	
3. Озвучивает необходимость перейти к подготовке презентации для защиты результатов проекта	Готовят презентацию для защиты результатов кейса
4. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 5	
Цель: Самостоятельная работа над кейсом (продолжение). Подготовка материалов для презентации результатов кейса	
Что делает педагог:	Что делают дети:
1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой. В дальнейшем следит за таймингом.	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.
2. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости. <i>Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на интересующую тему, или натолкнуть его на мысль, как получить результат на основе тех знаний, которые у него есть в данный момент.</i>	Самостоятельно работают над кейсом
3. Озвучивает необходимость перейти к подготовке презентации для защиты результатов проекта	Готовят презентацию для защиты результатов кейса
4. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 6	
Цель: Провести демонстрацию полученного за кейс результата, провести рефлекссию	
Что делает педагог:	Что делают дети:
1. Озвучивает цель занятия, определяет порядок демонстрации результатов кейса. Например, на выступление каждой команды не более 5-и минут, далее в течение 3-х минут вопросы.	

2. Устанавливает порядок, в котором учащиеся будут демонстрировать свои результаты. Например, по убыванию суммы букв в имени (и фамилии) участников команд.	Команды считают сумму букв в имени (и фамилии) участников и определяют порядок выступления.
3. Осуществляет модерацию презентаций результатов кейса участниками.	Участники / команда презентуют результаты кейса
4. Проводит рефлексию (лучше проводить, сидя в кругу). Задаёт вопросы учащимся: 1. Расскажите, чему вы научились за этот кейс, какие навыки освоили. 2. Удалось ли применить полученные навыки еще где-то? . 3. Что в этом кейсе далось вам легко, а что было трудным для освоения? Как вы справились с этими трудностями?	Отвечают на вопросы преподавателя: 1. Чему научились за этот кейс, какие навыки освоили. 2. Удалось ли применить полученные навыки еще где-то? . 3. Что в этом кейсе далось легко, а что было трудным для освоения? Как справились с этими трудностями?

Минимально необходимый уровень входных компетенций:

- запуск интегрированной среды разработки IDLE, создание и запуск программы на языке Python в файлах с расширением *.py;
- ввод с клавиатуры, ввод на экран на языке программирования Python;
- создание переменных, присваивание переменным значений;
- условный оператор, операторы циклов.

Предполагаемые результаты обучающихся, формируемые навыки:

- закрепят создание разветвляющихся и циклических алгоритмов на языке Python;
- освоят создание вложенных циклических алгоритмов;
- закрепят понятия «функция», «параметры функции», «возвращаемое значение»;
- научатся создавать свои функции на языке программирования Python;
- освоят возможности модуля turtle.

Артефакты:

Консольное приложение, которое последовательно создает картину.

С клавиатуры пользователь вводит настройки (по желанию). Результаты выводятся на экран.

Процедура и форма выявления образовательного результата

Представление результатов образовательной деятельности пройдет в форме публичной презентации решений кейса командами и последующих ответов выступающих на вопросы наставника и других команд.

Необходимые материалы и оборудование

Для преподавателя: персональный компьютер, медиапроектор.

Для обучающихся: персональный компьютер на каждого, флеш-накопитель (по желанию), доступ в сеть Интернет.

Расходные материалы

Для обучающихся: лист бумаги, ручка.

Программное обеспечение

- Язык программирования Python 3. (<https://www.python.org/downloads/>)
- интегрированная среда разработки и обучения на языке Python IDLE (устанавливается вместе с языком программирования Python)

Литература:

1. «Черепашья графика» при помощи turtle, рисование при помощи алгоритма:

<https://gvard.github.io/py/turtle/>

2. Полный список возможностей модуля turtle:

<http://docs.python.org/3/library/turtle.html>.

Приложения:

Возможности модуля turtle

Настройка холста и исполнителя:

reset() – открыть холст() или сбросить параметры по умолчанию();

clear() – очистить холст(), не перемещая черепаху;

shape(style) – изменение внешнего вида исполнителя; для параметра style возможны значения:

- arrow;
- turtle;
- circle;
- square;
- triangle;
- classic.

shapesize(x,y) – задание размера исполнителя;

pensize(d) – задание толщины следа;

color("red") – задание цвета следа (по цвету следа меняет цвет и исполнитель).

Перемещение и рисование:

forward(n) | fd(n) – вперед (пиксели);

backward(n) | bk(n) – назад (пиксели);
right(α) | rt(α) – направо (градусы);
left(α) | lt(α) – налево (градусы);
goto(x,y) – переместить(координата_X, координата_Y);
home() – вернуть исполнителя домой(), в центр холста;
circle() – рисовать окружность(радиус) или дугу (радиус, угол). Если радиус > 0 , то дуга или круг рисуется против часовой стрелки, если радиус < 0 , то по часовой стрелке.
undo() – отменить действие();
speed() – ускорение исполнителя (0..10).

Управление пером:

pendown() | pd() | down() – опустить перо();
penup() | pu() | up() – поднять перо();
pensize() | width() – размер пера (пиксели, по умолчанию 1);
color() – цвет («цвет пера», «цвет заливки»), например,
color("red", "green"),
color((r1,g1,b1), (r2,g2,b2)),
color("#285078", "#a0c8f0");
pencolor() – цвет («цвет пера»), по умолчанию черный;
fillcolor() – цвет («цвет заливки»);
begin_fill() – начать заливку();
end_fill() – закончить заливку().

Исходный код программы для рисования правильного n-угольника

```
import turtle
```

```
# создание холста
```

```
polygon = turtle.Turtle()
```

```
# настройка исполнителя и пера
```

```
polygon.shape("turtle")
```

```
polygon.shapesize(3,3)
```

```
polygon.pensize(5)
```

```
polygon.color("red")
```

```
# ввод исходных данных
```

```
num_sides = int(input("Введите количество сторон многоугольника: "))
```

```
side_length = 70
```

```
angle = 360.0 / num_sides
```

```
# ввод исходных данных
```

```
for i in range(num_sides):
```

```
    polygon.forward(side_length)
```

```
    polygon.right(angle)
```

Кейс№ 6

«Черепашка, математика и случайные числа»

Текст кейса

В двух кейсах, над которыми вы работали до этого, пользователь вводил информацию в программу с клавиатуры, а программа выводила для него информацию на экран. Информация эта была текстовой. В этом кейсе, вы научитесь писать программы, которые будут рисовать!

Помните, какие большие и серьезные модули языка программирования Python вы уже добавляли ваши программы? Для чего они были предназначены? Модуль `random` – для работы со случайными числами. Модуль `time()` – для работы со временем. Модуль `math()` – для выполнения различных математических действий. В этом кейсе вам понадобится модуль с немного смешным названием – `turtle`. Как ни странно, но название этого модуля так и переводится – «черепашка». И эта черепашка умеет рисовать.

Именно с помощью нее, применяя знания математики, вы в этом кейсе и будете создавать удивительные рисунки: правильные, четкие, с внутренней логикой и хаотичные, спонтанные.

Люди давным-давно стали создавать орнаменты. Их вырезали на деревянных ручках оружия, вышивали на одежде и полотенцах. Орнаменты использовали, когда ткали ковры. Орнаментами украшали посуду и здания.

Попробуйте и вы придумать свой орнамент (он должен быть достаточно сложным) и украсить им какой-то предмет.

С другой стороны интересны и рисунки, которые создаются хаотично, без какой-либо последовательности.

Попробуйте создать и такой рисунок и что-то им украсить.

Чтобы пользователь чувствовал себя участником процесса создания орнамента и «случайного» рисунка продумайте процесс их создания так так, чтобы перед началом пользователь мог сделать «заказ». Например, вы можете предложить пользователю выбрать тип фигуры, которая будет использоваться в рисунке, или задать количество фигур определенного типа.

Описание продукта

Приложение для рисования орнамента и случайного рисунка на каких-то предметах Настройки по желанию вводятся с клавиатуры. Вывод осуществляется на экран.

Категория кейса: продвинутый

Вопросы к кейсу:

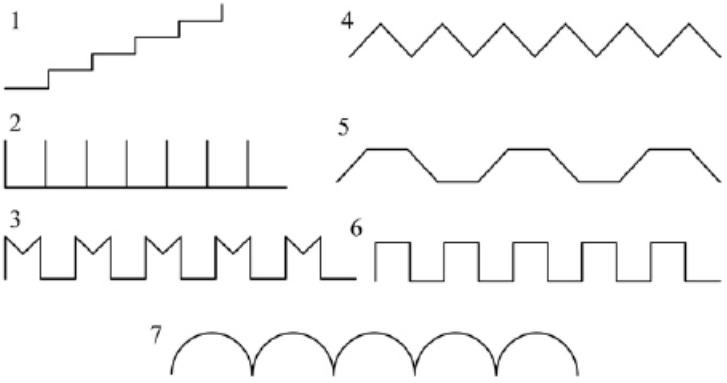
1. Вы уже не раз сталкивались с встроенными функциями языка программирования Python. Можете их перечислить?
2. Функции часто содержатся в модулях. Назовите несколько функций из модуля `math`, которые выполняют понятные вам действия.
3. Какие у функций особенности? Обязательно ли должны быть параметры у функции? Приведите примеры встроенных функций без параметров и функций, которые выполняют действия, но не возвращают результат.
4. А можно ли написать свою функцию на языке Python, а затем ее использовать? Каким образом это можно сделать? Как интерпретатор поймет, что вы собираетесь описывать функцию, где ее начало и конец. Как передать в функцию параметр, а затем вернуть результат?

Количество учебных часов/занятий, на которые рассчитан кейс: 12 ак. ч / 6 занятий

Учебно-тематическое планирование (занятие — 2 часа):

Занятие 1	
Цель: Познакомить обучающихся с возможностями модуля <code>turtle</code> . Уточнить понятия «функция», «параметры функции», «возвращаемое значение». Научить создавать свои функции на языке программирования Python.	
Что делает педагог:	Что делают дети:
1. Предлагает обучающимся немного поиграть перед началом работы над третьим кейсом. Игра "Что ты знаешь о черепахах" (на развитие навыков коммуникации). Участники делятся на 2 или 3 команды. В течение 5 минут, команды обсуждают, что они знают о черепахах. Это могут быть сведения о живых черепахах, памятниках, мультипликационных героях и т.д. (Можно фиксировать сведения на листе. Пользование интернетом может быть разрешено или запрещено.) Далее команды по очереди называют по одному факту о черепахах. Спикер от команды – не один, каждый раз от команды выступает новый участник. Выигрывает та команд, которая назовет факт последней.	Участвуют в игре.
2. Спрашивает или сам напоминает, чему научились обучающиеся во время работы над приветственным	Вспоминают, чему научились во время работы

и вводным кейсами. Знакомит обучающихся с третьим кейсом.	над первыми двумя кейсами.
3. Спрашивает, что нужно для рисования. Резюмирует: холст, кисть (или перо, если рисуем тушью).	Обучающиеся высказывают свои варианты.
4. Создает новый файл *.ру. Подключает модуль turtle. Создает холст. Задает настройки для холста и исполнителя.	Создает новый файл *.ру. Добавляют код в программу вслед за педагогом.
5. Знакомит обучающихся с основными командами для рисования пером (команды приведены в приложении к кейсу).	Добавляют код в программу вслед за педагогом.
6. Показывает, как нарисовать замкнутый контур и как его заполнить (команды приведены в приложении к кейсу) на примере квадрата и треугольника (пока без циклов). Перед тем как писать код желательно показать процесс рисования на флипчарте.	Добавляют код в программу вслед за педагогом.
7. Предлагает немного отвлечься и ответить на вопросы кейса. После ответов учащихся определяет понятия функция», «параметры функции», «возвращаемое значение».	Обучающиеся ищут ответы на вопросы кейса, используя ресурсы сети Интернет. Обсуждают.
8. Для повышения читабельности кода предлагает вынести рисование отдельных фигур в функции. После того как обучающие самостоятельно попробуют это сделать, реализует этот функционал.	Создают функции самостоятельно или воспроизводят код педагога.
9. Предлагает учащимся самостоятельно реализовать функции для рисования прямоугольника, ромба, трапеции. Углы можно определять транспортиром по рисунку или математически, если обучающиеся владеют соответствующим математическим аппаратом.	Сначала рисуют фигуру на листе бумаги, определяют углы поворота. Затем создают функции для рисования фигур. Запускают программу и отлаживают.
10. Предлагает учащимся собрать созданные функции в отдельный модуль (файл) и подключить его к своей программе.	Выносят все созданные функции в отдельный файл и подключают его к своей программе. Запускают программу и отлаживают.
17. Резюмирует, что было сделано на занятии.	Дополняют резюме преподавателя
18. Предлагает учащимся высказать свои впечатления от занятия.	Высказывают свои впечатления от занятия
Занятие 2	
Цель: Показать, как циклы могут использоваться для рисования. Познакомить учащихся с самостоятельным заданием на этот кейс. Планирование обучающимися работ над кейсом. Самостоятельная работа обучающихся над кейсом.	
Что делает педагог:	Что делают дети:
1. Предлагает обучающимся вспомнить, что было изучено на предыдущем занятии.	Вспоминают, что было изучено на предыдущем занятии.
2. Предлагает обучающимся использовать циклы для рисования простых последовательностей:	Создают новый файл *.ру. Создают программу по

	заданию: сначала без циклов, затем заменяют повторяющиеся строки циклами.
3. Предлагает обучающимся использовать циклы для рисования правильных многоугольников: равностороннего треугольника, квадрата, пятиугольника, шестиугольника. Рисование каждой фигуры желательно вынести в отдельную функцию. После того как обучающиеся попробуют сделать это сами, предлагает свое решение.	Создают новый файл *.ру. Создают функции по заданию. Запускают и отлаживают. При необходимости обращаются к коду педагога.
4. Предлагает обучающимся найти закономерности в рисовании правильных многоугольников. После обсуждения резюмирует. Реализует код функции для рисования правильного n-угольника.	Ищут закономерности в рисовании правильных многоугольников, обсуждают. Создают функцию для рисования правильного n-угольника.
5. Озвучивает кейс для самостоятельной работы и требования к нему.	Ищут и тестируют онлайн-тренажеры для сложения, умножения и других математических операций.
6. Модерирует процесс генерации идей для будущей картины.	Придумывают идеи для геометрического орнамента и «случайной» картины..
7. Озвучивает необходимость составления плана работ и назначения ответственных (на листе бумаге, на флипчарте с помощью стикеров, в программном обеспечении Trello или другом). Предлагает сформулировать, что будет входными данными, и что будет на выходе программы. Наблюдает за работой обучающихся, при необходимости консультирует.	Выбирают средство планирования, составляют план работ и назначают ответственных (при работе в паре).
8. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости.	Самостоятельно работают над кейсом.
9. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 3	
Цель: Самостоятельная работа над кейсом (продолжение).	

Что делает педагог:	Что делают дети:
1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой. В дальнейшем следит за таймингом.	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.
2. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости. <i>Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на интересующую тему, или натолкнуть его на мысль, как получить результат на основе тех знаний, которые у него есть в данный момент.</i>	Самостоятельно работают над кейсом
3. Озвучивает необходимость перейти к подготовке презентации для защиты результатов проекта	Готовят презентацию для защиты результатов кейса
4. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 4	
Цель: Самостоятельная работа над кейсом (продолжение)	
Что делает педагог:	Что делают дети:
1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой. В дальнейшем следит за таймингом.	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.
2. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости. <i>Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на</i>	Самостоятельно работают над кейсом

<i>интересующую тему, или натолкнуть его на мысль, как получить результат на основе тех знаний, которые у него есть в данный момент.</i>	
3. Озвучивает необходимость перейти к подготовке презентации для защиты результатов проекта	Готовят презентацию для защиты результатов кейса
4. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 5	
Цель: Самостоятельная работа над кейсом (продолжение). Подготовка материалов для презентации результатов кейса	
Что делает педагог:	Что делают дети:
1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой. В дальнейшем следит за таймингом.	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.
2. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости. <i>Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на интересующую тему, или натолкнуть его на мысль, как получить результат на основе тех знаний, которые у него есть в данный момент.</i>	Самостоятельно работают над кейсом
3. Озвучивает необходимость перейти к подготовке презентации для защиты результатов проекта	Готовят презентацию для защиты результатов кейса
4. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 6	
Цель: Провести демонстрацию полученного за кейс результата, провести рефлекссию	
Что делает педагог:	Что делают дети:
1. Озвучивает цель занятия, определяет порядок демонстрации результатов кейса. Например, на выступление каждой команды не более 5-и минут, далее в течение 3-х минут вопросы.	

2. Устанавливает порядок, в котором учащиеся будут демонстрировать свои результаты. Например, по убыванию суммы букв в имени (и фамилии) участников команд.	Команды считают сумму букв в имени (и фамилии) участников и определяют порядок выступления.
3. Осуществляет модерацию презентаций результатов кейса участниками.	Участники / команда презентуют результаты кейса
4. Проводит рефлексию (лучше проводить, сидя в кругу). Задает вопросы учащимся: 1. Расскажите, чему вы научились за этот кейс, какие навыки освоили. 2. Удалось ли применить полученные навыки еще где-то? 3. Что в этом кейсе далось вам легко, а что было трудным для освоения? Как вы справились с этими трудностями?	Отвечают на вопросы преподавателя: 1. Чему научились за этот кейс, какие навыки освоили. 2. Удалось ли применить полученные навыки еще где-то? . 3. Что в этом кейсе далось легко, а что было трудным для освоения? Как справились с этими трудностями?

Минимально необходимый уровень входных компетенций:

- запуск интегрированной среды разработки IDLE, создание и запуск программы на языке Python в файлах с расширением *.py;
- ввод с клавиатуры, вывод на экран на языке программирования Python;
- создание переменных, присваивание переменным значений;
- условный оператор, операторы циклов.

Предполагаемые результаты обучающихся, формируемые навыки:

- закрепят создание разветвляющихся и циклических алгоритмов на языке Python;
- освоят создание вложенных циклических алгоритмов;
- закрепят понятия «функция», «параметры функции», «возвращаемое значение»;
- научатся создавать свои функции на языке программирования Python;
- освоят возможности модуля turtle.

Артефакты:

Консольное приложение, которое рисует орнамент и «случайную» картину на каких-то предметах.

С клавиатуры пользователь вводит настройки (по желанию). Результаты выводятся на экран.

Процедура и форма выявления образовательного результата

Представление результатов образовательной деятельности пройдет в форме публичной презентации решений кейса командами и последующих ответов выступающих на вопросы наставника и других команд.

Необходимые материалы и оборудование

Для преподавателя: персональный компьютер, медиапроектор.

Для обучающихся: персональный компьютер на каждого, флеш-накопитель (по желанию), доступ в сеть Интернет.

Расходные материалы

Для обучающихся: лист бумаги, ручка.

Программное обеспечение

- Язык программирования Python 3. (<https://www.python.org/downloads/>)
- интегрированная среда разработки и обучения на языке Python IDLE (устанавливается вместе с языком программирования Python)

Литература:

1. «Черепашья графика» при помощи turtle, рисование при помощи алгоритма:

<https://gvard.github.io/py/turtle/>

1. Полный список возможностей модуля turtle:

<http://docs.python.org/3/library/turtle.html>.

Приложения:

Возможности модуля turtle

Настройка холста и исполнителя:

reset() – открыть холст() или сбросить параметры по умолчанию();

clear() – очистить холст(), не перемещая черепаху;

shape(style) – изменение внешнего вида исполнителя; для параметра style возможны значения:

- arrow;
- turtle;
- circle;
- square;
- triangle;
- classic.

shapesize(x,y) – задание размера исполнителя;

pensize(d) – задание толщины следа;

color("red") – задание цвета следа (по цвету следа меняет цвет и исполнитель).

Перемещение и рисование:

forward(n) | fd(n) – вперед (пиксели);

backward(n) | bk(n) – назад (пиксели);

right(α) | rt(α) – направо (градусы);

left(α) | lt(α) – налево (градусы);

goto(x,y) – переместить(координата_X, координата_Y);

home() – вернуть исполнителя домой(), в центр холста;

circle() – рисовать окружность(радиус) или дугу (радиус, угол). Если радиус > 0 , то дуга или круг рисуется против часовой стрелки, если радиус < 0 , то по часовой стрелке.

undo() – отменить действие();

speed() – ускорение исполнителя (0..10).

Управление пером:

pendown() | pd() | down() – опустить перо();

penup() | pu() | up() – поднять перо();

pensize() | width() – размер пера (пиксели, по умолчанию 1);

color() – цвет («цвет пера», «цвет заливки»), например,

color("red", "green"),

color((r1,g1,b1), (r2,g2,b2)),

color("#285078", "#a0c8f0");

pencolor() – цвет («цвет пера»), по умолчанию черный;

fillcolor() – цвет («цвет заливки»);

begin_fill() – начать заливку();

end_fill() – закончить заливку().

Исходный код программы для рисования правильного n-угольника

```
import turtle
```

```
# создание холста
```

```
polygon = turtle.Turtle()
```

```
# настройка исполнителя и пера
```

```
polygon.shape("turtle")  
polygon.shapesize(3,3)  
polygon.pensize(5)  
polygon.color("red")
```

```
# ввод исходных данных
```

```
num_sides = int(input("Введите количество сторон многоугольника: "))
```

```
side_length = 70
```

```
angle = 360.0 / num_sides
```

```
# ввод исходных данных
```

```
for i in range(num_sides):
```

```
    polygon.forward(side_length)
```

```
    polygon.right(angle)
```


Кейс № 7

«Поле чудес»

Текст кейса

Когда вы слышите «Поле чудес» – о чем думаете? Вспоминаете историю про Буратино? Или телевизионную программу, в которой нужно по буквам угадывать слова? Возможно, вы играли в такую игру в детстве на бумаге.

В этом кейсе вам предстоит создать свою программу, не телевизионную, а компьютерную, в которой тоже нужно будет по буквам угадывать слова.

Описание продукта

Консольная игра «Поле чудес», в которой нужно по буквам угадывать слова. Для отображения неугаданного слова используйте ASCII-графику – символы подчеркивания.

Для ввода и вывода информации используйте поясняющие надписи. Уведомляйте пользователя о старте и завершении программы.

Категория кейса: продвинутый

Вопросы к кейсу:

1. Что такое тип данных «список» в языке программирования Python?
2. Как присвоить переменной значение типа «список»? Как вывести список на экран?
3. Может ли список быть пустым? Как создать пустой список?
4. Значения каких типов данных могут входить в список? Обязательно ли тип данных должен быть одинаковым у всех элементов списка?
5. Как определить количество элементов в списке?
6. Как получить определенный элемент из списка? Как изменить определенный элемент в списке?
7. Как проверить, что определенный элемент входит в список.
8. Какие операции можно совершать над списками? Можно ли их складывать? Как добавить элемент в список и как удалить элемент из списка?
9. Что такое срез списка? Как получить несколько элементов из середины списка, с начала списка, с конца списка?
10. Как обойти список, используя циклы for и while?

Количество учебных часов/занятий, на которые рассчитан кейс: 15 ак. ч / 4

занятия

Учебно-тематическое планирование (занятие — 2 часа):

Занятие 1	
Цель: Познакомить обучающихся с работой со списками в языке программирования Python. Закрепить создание собственных функций, циклических алгоритмов на языке программирования Python.	
Что делает педагог:	Что делают дети:
1. Предлагает обучающимся немного поиграть перед началом работы над четвертым кейсом. Игра "В строй" (на развитие коммуникабельности и эмпатии, выявление лидера). Без слов. За определенное время (30с, 1 мин – в зависимости от численности группы). Выстроиться по алфавиту по первой букве имени, по последней букве имени, в обратном порядке по росту, по размеру ноги и т.д. Резюмировать, сколько заданий из максимального количества было выполнено. Обратить внимание на обучающихся, которые в процессе игры пытались организовывать процесс.	Участвуют в игре.
2. Спрашивает или сам напоминает, чему научились обучающиеся во время работы над предыдущими кейсами. Знакомит обучающихся с четвертым кейсом.	Вспоминают, чему научились во время работы над предыдущими кейсами.
3. Предлагает учащимся привести примеры списков. Создает новый файл *.ру. Показывает, как создать такие списки на Python и вывести их на экран. Предлагает обучающимся создать несколько списков самостоятельно, в том числе с разными типами данных. Резюмирует. Определяет понятие списка в Python.	Создает новый файл *.ру. Создают списки вслед за педагогом, самостоятельно создают списки.
5. На мини-примерах рассказывает о возможностях работы со списками в Python (по вопросам к кейсу). Предлагает по каждому вопросу реализовать свои варианты: 1) Может ли список быть пустым? Как создать пустой список? Значения каких типов данных могут входить в список? Обязательно ли тип данных должен быть одинаковым у всех элементов списка? 2) Как определить количество элементов в списке? 3) Как получить определенный элемент из списка? Как изменить определенный элемент в списке? 4) Как проверить, что определенный элемент входит в список. 5) Какие операции можно совершать над списками? Можно ли их складывать? Как добавить элемент в список и как удалить элемент из списка?	Создают новый файл *.ру. Добавляют код в программу вслед за педагогом. Реализуют свой похожий функционал

Что такое срез списка? Как получить несколько элементов из середины списка, с начала списка, с конца списка? Как обойти список, используя циклы for и while? По возможности разный функционал реализует в разных функциях – это позволит закрепить работу с функциями в Python и будет способствовать структурированию информации о списках.	
6. Резюмирует, что было сделано на занятии.	Дополняют резюме преподавателя.
7. Предлагает учащимся высказать свои впечатления от занятия.	Высказывают свои впечатления от занятия.
Занятие 2	
Цель: Углубить знания обучающихся по работе со списками в языке программирования Python. Закрепить создание собственных функций, циклических алгоритмов на языке программирования Python.	
Что делает педагог:	Что делают дети:
1. Спрашивает или сам напоминает, чему научились обучающиеся на прошлом занятии – что они теперь умеют делать со списками.	Вспоминают, чему на предыдущем занятии.
2. На мини-примерах рассказывает о возможностях работы со списками в Python (по вопросам к кейсу). Предлагает по каждому вопросу реализовать свои варианты: 1) Определяет понятие среза списка. 2) Как получить несколько элементов из середины списка, с начала списка, с конца списка? 3) Как обойти список, используя циклы for и while? 4) Как пройти по списку в обратном порядке? 5) Как отсортировать список? (функция sort) Как отсортировать список в обратном порядке (функция reversed()). По возможности разный функционал реализует в разных функциях – это позволит закрепить работу с функциями в Python и будет способствовать структурированию информации о списках.	Создают новый файл *.py. Добавляют код в программу вслед за педагогом. Реализуют свой похожий функционал
3. Резюмирует, что было сделано на занятии.	Дополняют резюме преподавателя.
4. Предлагает учащимся высказать свои впечатления от занятия.	Высказывают свои впечатления от занятия.
Занятие 3	
Цель: Познакомить учащихся с самостоятельным заданием на этот кейс. Планирование обучающимися работ над кейсом. Самостоятельная работа обучающихся над кейсом.	
Что делает педагог:	Что делают дети:
1. Озвучивает кейс для самостоятельной работы и требования к нему.	Ищут и тестируют онлайн-тренажеры для сложения,

	умножения и других математических операций.
2. Предлагает учащимся распределиться на команды по 2 – 3 человека.	Распределяются на команды.
3. Модерирует процесс генерации идей.	Придумывают идею.
4. Озвучивает необходимость составления плана работ и назначения ответственных (на листе бумаге, на флипчарте с помощью стикеров, в программном обеспечении Trello или другом). Предлагает сформулировать, что будет входными данными, и что будет на выходе программы. Наблюдает за работой обучающихся, при необходимости консультирует.	Выбирают средство планирования, составляют план работ и назначают ответственных (при работе в паре).
5. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости.	Самостоятельно работают над кейсом.
6. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 3	
Цель: Самостоятельная работа над кейсом (продолжение).	
Что делает педагог:	Что делают дети:
1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой. В дальнейшем следит за таймингом.	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.
2. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости. <i>Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на интересующую тему, или натолкнуть его на мысль, как получить результат на основе тех знаний, которые у него есть в данный момент.</i>	Самостоятельно работают над кейсом
3. Озвучивает необходимость перейти к подготовке презентации для защиты результатов проекта	Готовят презентацию для защиты результатов кейса
4. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный

	момент сложностях. Ищут пути решения.
Занятие 4	
Цель: Самостоятельная работа над кейсом (продолжение)	
Что делает педагог:	Что делают дети:
1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой. В дальнейшем следит за таймингом.	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.
2. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости. <i>Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на интересующую тему, или натолкнуть его на мысль, как получить результат на основе тех знаний, которые у него есть в данный момент.</i>	Самостоятельно работают над кейсом
3. Озвучивает необходимость перейти к подготовке презентации для защиты результатов проекта	Готовят презентацию для защиты результатов кейса
4. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 5	
Цель: Самостоятельная работа над кейсом (продолжение)	
Что делает педагог:	Что делают дети:
1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой. В дальнейшем следит за таймингом.	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.

2. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости. <i>Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на интересующую тему, или натолкнуть его на мысль, как получить результат на основе тех знаний, которые у него есть в данный момент.</i>	Самостоятельно работают над кейсом
Занятие 6	
Цель: Самостоятельная работа над кейсом (продолжение). Подготовка материалов для презентации результатов кейса	
Что делает педагог:	Что делают дети:
1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой. В дальнейшем следит за таймингом.	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.
2. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости. <i>Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на интересующую тему, или натолкнуть его на мысль, как получить результат на основе тех знаний, которые у него есть в данный момент.</i>	Самостоятельно работают над кейсом
3. Озвучивает необходимость перейти к подготовке презентации для защиты результатов проекта	Готовят презентацию для защиты результатов кейса
4. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 7	
Цель: Провести демонстрацию полученного за кейс результата	
Что делает педагог:	Что делают дети:
1. Озвучивает цель занятия, определяет порядок демонстрации результатов кейса. Например, на выступление каждой команды не более 5-и минут, далее в течение 3-х минут вопросы.	

2. Устанавливает порядок, в котором учащиеся будут демонстрировать свои результаты. Например, по убыванию суммы букв в имени (и фамилии) участников команд.	Команды считают сумму букв в имени (и фамилии) участников и определяют порядок выступления.
3. Осуществляет модерацию презентаций результатов кейса участниками.	Участники / команда презентуют результаты кейса
4. Подводит количественные и качественные итоги по курсу.	
Занятие 8	
Цель: Провести рефлексию продвинутого кейса и всего курса	
Что делает педагог:	Что делают дети:
1. Проводит рефлексию кейса (лучше проводить, сидя в кругу). Задает вопросы учащимся: 1. Расскажите, чему вы научились за этот кейс, какие навыки освоили. 2. Удалось ли применить полученные навыки еще где-то? 3. Что в этом кейсе далось вам легко, а что было трудным для освоения? Как вы справились с этими трудностями?	Отвечают на вопросы преподавателя.
2. Проводит рефлексию курса. Задает вопросы учащимся: 1. Насколько вам было комфортно в течение этого курса? Что доставляло неудобство / беспокойство? Как Вы с этим справлялись? 2. Как оцениваете свои успехи за этот курс (можно по шкале от 1 до 5, или от 1 до 10)? Помните, какие цели вы ставили себе на этот курс? Удалось ли их достичь? 3. Какие ваши действия помогли вам добиться успеха? А какие мешали? 4. Что бы вы еще хотели научиться делать с помощью языка Python? Как вы планируете это делать?	Отвечают на вопросы преподавателя.
3. Подводит итоги.	
4. Предлагает участникам дать обратную связь по курсу в свободной форме.	Высказываются по желанию.

Минимально необходимый уровень входных компетенций:

- запуск интегрированной среды разработки IDLE, создание и запуск программы на языке Python в файлах с расширением *.py;
- ввод с клавиатуры, ввод на экран на языке программирования Python;
- создание переменных, присваивание переменным значений;
- условный оператор, операторы циклов, создание функций.

Предполагаемые результаты обучающихся, формируемые навыки:

- закрепят создание разветвляющихся и циклических алгоритмов на языке Python, создание функций;

- освоят работу со списками;
- познакомятся с ASCII-графикой.

Артефакты:

Консольная игра по отгадывания слов по буквам.

Процедура и форма выявления образовательного результата

Представление результатов образовательной деятельности пройдет в форме публичной презентации решений кейса командами и последующих ответов выступающих на вопросы наставника и других команд.

Необходимые материалы и оборудование

Для преподавателя: персональный компьютер, медиапроектор.

Для обучающихся: персональный компьютер на каждого, флеш-накопитель (по желанию), доступ в сеть Интернет.

Расходные материалы

Для обучающихся: лист бумаги, ручка.

Программное обеспечение

- Язык программирования Python 3. (<https://www.python.org/downloads/>)
- интегрированная среда разработки и обучения на языке Python IDLE (устанавливается вместе с языком программирования Python)

Литература:

1. Списки в языке программирования Python:

<https://younglinux.info/python/list>

Приложения:

Исходный код игры «Поле чудес»

```
import random
wordsAnimals = ['аист', 'бабуин', 'волк', 'голубь', 'жаба', 'зебра', 'индюк', 'кит', 'кобра',
'ласка', 'моль', 'носорог', 'олень', 'питон', 'сунс', 'тюлень', 'утка', 'форель', 'хорек',
'черепаха', 'ящерица']

def getRandomWord(wordList):
    # Функция возвращает случайную строку из переданного списка.
    wordIndex = random.randint(0, len(wordList) - 1)
    return wordList[wordIndex]

def displayBoard(missedLetters, correctLetters, secretWord):
    print('Ошибочные буквы:', end=' ')
    for letter in missedLetters:
        print(letter, end=' ')
```



```

print()

blanks = '_' * len(secretWord)

for i in range(len(secretWord)): # заменяет пропуски отгаданными буквами
    if secretWord[i] in correctLetters:
        blanks = blanks[:i] + secretWord[i] + blanks[i+1:]

for letter in blanks: # Показывает секретное слово с пробелами между буквами
    print(letter, end=' ')
print()

def getGuess(alreadyGuessed):
    # Функция возвращает букву, введенную игроком. Также проверяет, что игрок ввел
    # только одну букву и ничего больше.
    while True:
        print('Введите букву.')
        guess = input()
        guess = guess.lower()
        if len(guess) != 1:
            print('Нужно ввести только одну букву.')
        elif guess in alreadyGuessed:
            print('Ты уже называл эту букву. Назови другую.')
        elif guess not in 'абвгдежзийклмнопрстуфхцчшщъыьэюя':
            print('Пожалуйста, введи БУКВУ.')
        else:
            return guess

def playAgain():
    # Функция возвращает значение True, если игрок хочет сыграть заново; в противном
    # случае возвращает False.
    print('Хочешь сыграть еще раз? (Введи да или нет)')
    if playAgain == 'да' or playAgain == 'д':
        return True
    else:
        return False

# Основная программа
print('П О Л Е Ч У Д Е С')
print('Угадай животное')
missedLetters = ''
correctLetters = ''
secretWord = getRandomWord(wordsAnimals)
gameIsDone = False
max = 25

while True:
    displayBoard(missedLetters, correctLetters, secretWord)

    # игроку вводит букву.

```

```

guess = getGuess(missedLetters + correctLetters)
if guess in secretWord:
    correctLetters = correctLetters + guess
    # проверяем, выиграл ли игрок.
    foundAllLetters = True
    for i in range(len(secretWord)):
        if secretWord[i] not in correctLetters:
            foundAllLetters = False
            break
    if foundAllLetters:
        print('ДА! Секретное слово - "' + secretWord + '"! Ты угадал!')
        gameIsDone = True
else:
    missedLetters = missedLetters + guess

    # проверяем, превысил ли игрок лимит попыток и проиграл.
    if len(missedLetters) == max:
        displayBoard(missedLetters, correctLetters, secretWord)
        print('Увы, ты использовал все попытки!\nНе угадано букв: ' +
str(len(missedLetters)) + ' и угадано букв: ' + str(len(correctLetters)) + '. Было загадано
слово "' + secretWord + '".')
        gameIsDone = True

# спрашиваем, хочет ли игрок сыграть заново (только если игра завершена)
if gameIsDone:
    if playAgain():
        missedLetters = ""
        correctLetters = ""
        gameIsDone = False
        secretWord = getRandomWord(wordsAnimals)
    else:
        break

```

Кейс № 8

«Сам себе чертежник»

Текст кейса

«Черепашья графика» – довольно занятная штука. А если кто-то не знает язык программирования Python? Как быть? Оставить всякие надежды познакомиться с черепашьей графикой? Совсем нет!

Ваших знаний почти достаточно, чтобы сделать программу, которая будет рисовать то, что хочет пользователь: пользователь вводит команды, черепашка рисует. Чтобы код вашей программы получился лаконичным и красивым, предлагаем в этом кейсе познакомиться со списками и активно использовать их в коде вашей программы. Списки – очень полезная вещь и встречаются повсеместно. Вот несколько примеров.

Мама отправляет вас в магазин и вручает листок. На нем – список того, что нужно купить. Три, пять, а может быть даже десять позиций просит вас мама купить в магазине. Это список.

Посмотрите на алфавит, буквы выстроены в ряд. Это тоже список.

Посмотрите на какое-нибудь слово. Можете назвать, какая буква стоит в начале, какая буква вторая по счету, а какая в конце? Каждое слово – это тоже список, список букв.

Почти во всех языках программирования есть специальные типы данных, которые позволяют хранить последовательности и работать с ними. В языке программирования Python этот тип данных называется список (в других языках программирования он также может называться массивом).

Описание продукта

Консольное приложение, которое создает черепашью графику по набору команд, вводимых пользователем.

Категория кейса: продвинутый

Вопросы к кейсу:

1. Что такое тип данных «список» в языке программирования Python?
2. Как присвоить переменной значение типа «список»? Как вывести список на экран?
3. Может ли список быть пустым? Как создать пустой список?

4. Значения каких типов данных могут входить в список? Обязательно ли тип данных должен быть одинаковым у всех элементов списка?
5. Как определить количество элементов в списке?
6. Как получить определенный элемент из списка? Как изменить определенный элемент в списке?
7. Как проверить, что определенный элемент входит в список.
8. Какие операции можно совершать над списками? Можно ли их складывать? Как добавить элемент в список и как удалить элемент из списка?
9. Что такое срез списка? Как получить несколько элементов из середины списка, с начала списка, с конца списка?
10. Как обойти список, используя циклы for и while?

Количество учебных часов/занятий, на которые рассчитан кейс: 15 ак. ч / 7,5 занятий

Учебно-тематическое планирование (занятие — 2 часа):

Занятие 1	
Цель: Познакомить обучающихся с работой со списками в языке программирования Python. Закрепить создание собственных функций, циклических алгоритмов на языке программирования Python.	
Что делает педагог:	Что делают дети:
1. Предлагает обучающимся немного поиграть перед началом работы над четвертым кейсом. Игра "В строй" (на развитие коммуникабельности и эмпатии, выявление лидера). Без слов. За определенное время (30с, 1 мин – в зависимости от численности группы). Выстроиться по алфавиту по первой букве имени, по последней букве имени, в обратном порядке по росту, по размеру ноги и т.д. Резюмировать, сколько заданий из максимального количества было выполнено. Обратить внимание на обучающихся, которые в процессе игры пытались организовывать процесс.	Участвуют в игре.
2. Спрашивает или сам напоминает, чему научились обучающиеся во время работы над предыдущими кейсами. Знакомит обучающихся с четвертым кейсом.	Вспоминают, чему научились во время работы над предыдущими кейсами.
3. Предлагает обучающимся привести примеры списков. Создает новый файл *.py. Показывает, как создать такие списки на Python и вывести их на экран. Предлагает обучающимся создать несколько списков самостоятельно, в том числе с разными типами данных. Резюмирует. Определяет понятие списка в Python.	Создает новый файл *.py. Создают списки вслед за педагогом, самостоятельно создают списки.

5. На мини-примерах рассказывает о возможностях работы со списками в Python (по вопросам к кейсу). Предлагает по каждому вопросу реализовать свои варианты: 1) Может ли список быть пустым? Как создать пустой список? Значения каких типов данных могут входить в список? Обязательно ли тип данных должен быть одинаковым у всех элементов списка? 2) Как определить количество элементов в списке? 3) Как получить определенный элемент из списка? Как изменить определенный элемент в списке? 4) Как проверить, что определенный элемент входит в список. 5) Какие операции можно совершать над списками? Можно ли их складывать? Как добавить элемент в список и как удалить элемент из списка? Что такое срез списка? Как получить несколько элементов из середины списка, с начала списка, с конца списка? Как обойти список, используя циклы for и while? По возможности разный функционал реализует в разных функциях – это позволит закрепить работу с функциями в Python и будет способствовать структурированию информации о списках.	Создают новый файл *.py. Добавляют код в программу вслед за педагогом. Реализуют свой похожий функционал
6. Резюмирует, что было сделано на занятии.	Дополняют резюме преподавателя.
7. Предлагает учащимся высказать свои впечатления от занятия.	Высказывают свои впечатления от занятия.
Занятие 2	
Цель: Углубить знания обучающихся по работе со списками в языке программирования Python. Закрепить создание собственных функций, циклических алгоритмов на языке программирования Python.	
Что делает педагог:	Что делают дети:
1. Спрашивает или сам напоминает, чему научились обучающиеся на прошлом занятии – что они теперь умеют делать со списками.	Вспоминают, чему на предыдущем занятии.
2. На мини-примерах рассказывает о возможностях работы со списками в Python (по вопросам к кейсу). Предлагает по каждому вопросу реализовать свои варианты: 1) Определяет понятие среза списка. 2) Как получить несколько элементов из середины списка, с начала списка, с конца списка? 3) Как обойти список, используя циклы for и while? 4) Как пройти по списку в обратном порядке? 5) Как отсортировать список? (функция sort) Как отсортировать список в обратном порядке (функция reversed()).	Создают новый файл *.py. Добавляют код в программу вслед за педагогом. Реализуют свой похожий функционал

По возможности разный функционал реализует в разных функциях – это позволит закрепить работу с функциями в Python и будет способствовать структурированию информации о списках.	
3. Резюмирует, что было сделано на занятии.	Дополняют резюме преподавателя.
4. Предлагает учащимся высказать свои впечатления от занятия.	Высказывают свои впечатления от занятия.
Занятие 3	
Цель: Познакомить учащихся с самостоятельным заданием на этот кейс. Планирование обучающимися работ над кейсом. Самостоятельная работа обучающихся над кейсом.	
Что делает педагог:	Что делают дети:
1. Озвучивает кейс для самостоятельной работы и требования к нему.	Ищут и тестируют онлайн-тренажеры для сложения, умножения и других математических операций.
2. Предлагает учащимся распределиться на команды по 2 – 3 человека.	Распределяются на команды.
3. Модерирует процесс генерации идей.	Придумывают идею.
4. Озвучивает необходимость составления плана работ и назначения ответственных (на листе бумаге, на флипчарте с помощью стикеров, в программном обеспечении Trello или другом). Предлагает сформулировать, что будет входными данными, и что будет на выходе программы. Наблюдает за работой обучающихся, при необходимости консультирует.	Выбирают средство планирования, составляют план работ и назначают ответственных (при работе в паре).
5. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости.	Самостоятельно работают над кейсом.
6. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 3	
Цель: Самостоятельная работа над кейсом (продолжение).	
Что делает педагог:	Что делают дети:
1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой /	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.

командой. В дальнейшем следит за таймингом.	
2. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости. <i>Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на интересующую тему, или натолкнуть его на мысль, как получить результат на основе тех знаний, которые у него есть в данный момент.</i>	Самостоятельно работают над кейсом
3. Озвучивает необходимость перейти к подготовке презентации для защиты результатов проекта	Готовят презентацию для защиты результатов кейса
4. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 4	
Цель: Самостоятельная работа над кейсом (продолжение)	
Что делает педагог:	Что делают дети:
1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой. В дальнейшем следит за таймингом.	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.
2. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости. <i>Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на интересующую тему, или натолкнуть его на мысль, как получить результат на основе тех знаний, которые у него есть в данный момент.</i>	Самостоятельно работают над кейсом
3. Озвучивает необходимость перейти к подготовке презентации для защиты результатов проекта	Готовят презентацию для защиты результатов кейса
4. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.

Занятие 5	
Цель: Самостоятельная работа над кейсом (продолжение)	
Что делает педагог:	Что делают дети:
1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой. В дальнейшем следит за таймингом.	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.
2. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости. <i>Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на интересующую тему, или натолкнуть его на мысль, как получить результат на основе тех знаний, которые у него есть в данный момент.</i>	Самостоятельно работают над кейсом
Занятие 6	
Цель: Самостоятельная работа над кейсом (продолжение). Подготовка материалов для презентации результатов кейса	
Что делает педагог:	Что делают дети:
1. Озвучивает цель и план занятия: - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой; - самостоятельная работа над кейсом, консультации преподавателя; - подготовка презентации для защиты результатов кейса; - обсуждение положения дел и имеющихся сложностей на настоящий момент с каждым участником / парой / командой. В дальнейшем следит за таймингом.	Вспоминают, чем закончилось предыдущее занятие, рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях.
2. Наблюдает за продвижением участников / пар / команд, консультирует по мере необходимости. <i>Если участник задает вопрос по теме, которая будет рассматриваться в дальнейшем, в зависимости от уровня и мотивации участника целесообразно или помочь участнику с подбором материалов на интересующую тему, или натолкнуть его на мысль, как получить результат на основе тех знаний, которые у него есть в данный момент.</i>	Самостоятельно работают над кейсом

3. Озвучивает необходимость перейти к подготовке презентации для защиты результатов проекта	Готовят презентацию для защиты результатов кейса
4. Предлагает учащимся обсудить положение дел, консультирует по возможным трудностям и способам их преодоления до следующего занятия.	Рассказывают о положении дел по решению кейса и имеющихся на данный момент сложностях. Ищут пути решения.
Занятие 7	
Цель: Провести демонстрацию полученного за кейс результата	
Что делает педагог:	Что делают дети:
1. Озвучивает цель занятия, определяет порядок демонстрации результатов кейса. Например, на выступление каждой команды не более 5-и минут, далее в течение 3-х минут вопросы.	
2. Устанавливает порядок, в котором учащиеся будут демонстрировать свои результаты. Например, по убыванию суммы букв в имени (и фамилии) участников команд.	Команды считают сумму букв в имени (и фамилии) участников и определяют порядок выступления.
3. Осуществляет модерацию презентаций результатов кейса участниками.	Участники / команда презентуют результаты кейса
4. Подводит количественные и качественные итоги по курсу.	
Занятие 8	
Цель: Провести рефлексию продвинутого кейса и всего курса	
Что делает педагог:	Что делают дети:
1. Проводит рефлексию кейса (лучше проводить, сидя в кругу). Задает вопросы учащимся: 1. Расскажите, чему вы научились за этот кейс, какие навыки освоили. 2. Удалось ли применить полученные навыки еще где-то? 3. Что в этом кейсе далось вам легко, а что было трудным для освоения? Как вы справились с этими трудностями?	Отвечают на вопросы преподавателя.
2. Проводит рефлексию курса. Задает вопросы учащимся: 1. Насколько вам было комфортно в течение этого курса? Что доставляло неудобство / беспокойство? Как Вы с этим справлялись? 2. Как оцениваете свои успехи за этот курс (можно по шкале от 1 до 5, или от 1 до 10)? Помните, какие цели вы ставили себе на этот курс? Удалось ли их достичь? 3. Какие ваши действия помогли вам добиться успеха? А какие мешали? 4. Что бы вы еще хотели научиться делать с помощью языка Python? Как вы планируете это делать?	Отвечают на вопросы преподавателя.
3. Подводит итоги.	
4. Предлагает участникам дать обратную связь по курсу в свободной форме.	Высказываются по желанию.

Минимально необходимый уровень входных компетенций:

- запуск интегрированной среды разработки IDLE, создание и запуск программы на языке Python в файлах с расширением *.py;
- ввод с клавиатуры, вывод на экран на языке программирования Python;
- создание переменных, присваивание переменным значений;
- условный оператор, операторы циклов, создание функций.

Предполагаемые результаты обучающихся, формируемые навыки:

- закрепят создание разветвляющихся и циклических алгоритмов на языке Python, создание функций;
- освоят работу со списками;
- освоят новые возможности модуля turtle.

Артефакты:

Консольное приложение, которое создает черепашую графику по набору команд, вводимых пользователем.

Процедура и форма выявления образовательного результата

Представление результатов образовательной деятельности пройдет в форме публичной презентации решений кейса командами и последующих ответов выступающих на вопросы наставника и других команд.

Необходимые материалы и оборудование

Для преподавателя: персональный компьютер, медиапроектор.

Для обучающихся: персональный компьютер на каждого, флеш-накопитель (по желанию), доступ в сеть Интернет.

Расходные материалы

Для обучающихся: лист бумаги, ручка.

Программное обеспечение

- Язык программирования Python 3. (<https://www.python.org/downloads/>)
- интегрированная среда разработки и обучения на языке Python IDLE (устанавливается вместе с языком программирования Python)

Литература:

2. Списки в языке программирования Python:

<https://younglinux.info/python/list>

3. Черепашня графика» при помощи turtle, рисование при помощи алгоритма:

<https://gvard.github.io/py/turtle/>

4. Полный список возможностей модуля turtle:

<http://docs.python.org/3/library/turtle.html>.

Приложения:

Исходный код варианта приложения «Сам себе чертежник»

```
good_commands = ['F', 'B', 'R', 'L', 'U', 'D', 'N']

from turtle import *

def turtle_controller(do, val):
    do = do.upper()
    if do == 'F':
        forward(val)
    elif do == 'B':
        backward(val)
    elif do == 'R':
        right(val)
    elif do == 'L':
        left(val)
    elif do == 'U':
        penup(val)
    elif do == 'D':
        pendown(val)
    elif do == 'N':
        reset(val)
    else:
        print(do, ' - ошибочная команда')

def do_commands(program):
    cmd_list = program.split('-')
    for command in cmd_list:
        #проверяем, что команда не пустая
        cmd_len = len(command)
        if cmd_len == 0:
            continue

        # первый символ команды - направление или действие
        cmd_type = command[0]

        if cmd_type not in good_commands:
            print (cmd_type, " - ошибочная команда!")
            break

        # остальные символы команды - числовое значение
        num = 0
        if cmd_len > 1:
            num_string = command[1:]
            num = int(num_string)
```

```
# определили, что нужно сделать, отдаем команду на выполнение
turtle_controller(cmd_type, num)

instructions = "Введите программу для черепашки: "
screen = getscreen()
while True:
    # формируем окно для пользовательского ввода
    t_program = screen.textinput('Drawing Machine', instructions)
    if t_program == None or t_program.upper() == 'END':
        break

do_commands(t_program)
```