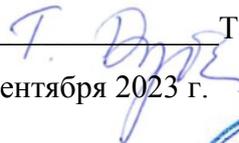


АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«УЧЕБНЫЙ ЦЕНТР СКБ КОНТУР»

Утверждаю
Директор АНО ДПО
«Учебный центр СКБ Контур»


Т.В. Рубан
1 сентября 2023 г.



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА

дополнительная общеразвивающая программа

ОСНОВЫ ЯЗЫКА РУТНОН

Документ: Свидетельство

Форма обучения: заочная с использованием дистанционных образовательных технологий, электронного обучения

Срок обучения: 3 недели

Объем программы: 38 ак. ч.

Москва, 2023 г.

Оглавление

1. ОБЩИЕ ПОЛОЖЕНИЯ.....	3
1.1. Назначение программы.....	3
1.2. Нормативные документы, регламентирующие разработку дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2. ОБЩАЯ ХАРАКТЕРИСТИКА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	3
2.1. Цели дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2.2. Требования к слушателю дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2.3. Срок освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2.4. Трудоемкость дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	3
2.5. Планируемые результаты обучения по итогам освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	4
3. СОДЕРЖАНИЕ И ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА.....	5
3.1. Учебный план.....	5
3.2. Содержание.....	7
3.3 Календарный учебный график.....	13
4. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ - ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ.....	14
4.1. Требования к квалификации педагогических кадров, представителей предприятий и организаций, обеспечивающих реализацию образовательного процесса.....	14
4.2. Требования к материально-техническим условиям.....	14
4.3. Информационные и учебно-методические условия реализации дополнительной общеобразовательной программы - дополнительной общеразвивающей программы.....	16
4.4. Общие требования к организации образовательного процесса.....	17
5. СИСТЕМА ОЦЕНКИ КАЧЕСТВА ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	18
5.1. Формы контроля знаний и требования к его проведению.....	18
5.2. Критерии оценки знаний слушателей.....	19
5.3. Фонд оценочных средств.....	20

1. ОБЩИЕ ПОЛОЖЕНИЯ

1.1. Назначение программы

Дополнительная общеобразовательная программа дополнительная общеразвивающая программа «Основы языка Python» направлена на получение практических навыков и теоретических знаний, необходимых для тех, кто хочет освоить навыки по основам программирования на языке Python.

Программа регламентирует цели, планируемые результаты, содержание, условия и технологии реализации образовательного процесса, оценку качества подготовки слушателей и включает в себя: учебный план, фонды оценочных средств, программу итоговой аттестации, календарный учебный график и другие материалы, обеспечивающие качество подготовки слушателей.

1.2. Нормативные документы, регламентирующие разработку дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

1. Трудовой кодекс Российской Федерации от 30.12.2001 № 197-ФЗ.
2. Федеральный закон Российской Федерации от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации».
3. Приказ Минпросвещения РФ от 27.07.2022 №629 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам".

2. ОБЩАЯ ХАРАКТЕРИСТИКА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

2.1. Цели дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Основная цель программы – дать слушателям знания и навыки по основам программирования на языке Python.

Задачи дисциплины

В процессе изучения курса «Основы языка Python» необходимо решить следующие задачи и рассказать слушателям:

- об основах написания программы на языке Python;
- об эффективном использовании библиотеки языка для повышения производительности кода;
- о том, как работать с данными с помощью языка программирования Python;
- о том, как создавать код с обработкой исключений;
- о работе в парадигме объектно-ориентированного программирования.

2.2. Требования к слушателю дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

К освоению дополнительной общеобразовательной программы – дополнительной общеразвивающей программы допускаются любые лица без предъявления требований к уровню образования.

Возраст слушателей: 18 лет и старше.

2.3. Срок освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Нормативный срок освоения программы – 3 недели.

2.4. Трудоемкость дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Объем образовательной программы составляет 38 академических часов. Из них 24 часа - самостоятельная работа, 14 академических часов – работа на образовательной онлайн-платформе.

2.5. Планируемые результаты обучения по итогам освоения дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

В результате обучения слушатели должны будут овладеть следующими знаниями, умениями и навыками:

Знать:

- особенности применения языка программирования Python;
- основы языка программирования Python;
- основные конструкции языка;
- особенности обработки исключений.

Уметь:

- писать программы на языке Python используя различные типы данных;
- читать код, созданный на языке программирования Python.

Владеть:

- навыками работы со словарями, коллекциями, библиотеками в Python;
- навыками обработки исключений.

3. СОДЕРЖАНИЕ И ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

3.1. Учебный план для заочной формы обучения с ДОТ и ЭО

№ п/п	Наименование разделов, дисциплин	Всего часов	Из них:		Форма контроля
			Самостоятельн ая работа	Работа на образовател ьной онлайн- платформе	
1.	Типы и структуры данных	3	2	1	Зачет
1.1	Типы и структуры данных	3	2	1	Тестирован ие
2.	Основные конструкции языка	6	4	2	Зачет
2.1	Основные конструкции языка	6	4	2	Тестирован ие
3.	Строки	3	2	1	Зачет
3.1	Работа со строками	3	2	1	Тестирован ие
4.	Списки	3	2	1	Зачет
4.1	Работа со списками	3	2	1	Тестирован ие
5.	Словари и коллекции	3	2	1	Зачет
5.1	Работа со словарями и коллекциями	3	2	1	Тестирован ие
6.	Функции	3	2	1	Зачет
6.1	Использование функций	3	2	1	Тестирован ие
7.	Классы и объекты	3	2	1	Зачет
7.1	Классы и объекты	3	2	1	Тестирован ие
8.	Объектно-ориентированное программирование	3	2	1	Зачет
8.1	Принципы ООП	3	2	1	Тестирован ие
9.	Инструменты инкапсуляции в Python	3	2	1	Зачет
9.1	Модификаторы доступа, свойства и дескрипторы	3	2	1	Тестирован ие
10.	Библиотеки	3	2	1	Зачет

10.1	Использование библиотек	3	2	1	Тестирование
11.	Исключения в Python	3	2	1	Зачет
11.1	Обработка исключений	3	2	1	Тестирование
ИТОГОВАЯ АТТЕСТАЦИЯ		2	–	2	Зачет
Всего:		38	24	14	–

3.2. Содержание

Тема 1. Типы и структуры данных

1.1. Типы и структуры данных

- Типы данных в языке Python
 - Вступление
 - Числовые типы данных
 - Целые
 - Числа с плавающей точкой
 - Комплексные числа
 - Строки
 - Типы данных логических значений
 - Шпаргалка по типам
- Структуры данных в языке Python
 - Вступление
 - Списки
 - Кортежи
 - Словари
 - Множества
 - Шпаргалка по структурам данных
- Практическое задание.

Задание 1. Напишите программу, которая запрашивает у пользователя его имя, возраст, рост и вес, а затем выводит на экран информацию о пользователе в следующем формате:

```
"Привет, [имя]! Твой возраст [возраст] лет, твой рост [рост] см, твой вес [вес] кг."
```

При написании программы используйте различные типы данных, такие как строки, целые и дробные числа. Не забудьте о преобразовании типов данных, если это необходимо.

Задание 2. Напишите программу, которая принимает на вход список из 10 случайных целых чисел в диапазоне от 1 до 100 и использует алгоритм сортировки (например, сортировку пузырьком), чтобы отсортировать этот список в порядке возрастания.

Затем программа должна вывести отсортированный список на экран.

Пример входных данных:

```
[56, 22, 87, 19, 4, 75, 36, 91, 38, 50]
```

Пример выходных данных:

```
[4, 19, 22, 36, 38, 50, 56, 75, 87, 91]
```

Тема 2. Основные конструкции языка

1.2. Основные конструкции языка

- Переменные, константы и оператор присваивания в Python
 - Введение
 - Переменные
 - Константы
 - Оператор присваивания
- Условный оператор в языке Python
 - Условный оператор if
 - Условный оператор if-else
 - Условный оператор if-elif-else
 - Оператор условного выражения (тернарный оператор)
- Циклы в языке Python
 - Цикл for
 - Цикл while

- Практическое задание.

Задание 1.

1. Создайте программу на Python, которая будет вычислять площадь круга.
2. Задайте значение радиуса круга в переменную и используйте оператор присваивания для присвоения значения.
3. Используйте константу для хранения значения числа π .
4. Используйте формулу $S = \pi r^2$ для вычисления площади круга, где S — площадь, r — радиус.
5. Выведите результат на экран с помощью функции `print()`.

Задание 2. Напишите функцию `sale_pirojki`, функция принимает 1 параметр: n .

n — количество пирожков, которые покупатель купит. Разные числа имеют разные цены (см. следующую таблицу). Возвращает, сколько денег покупатель потратит, чтобы купить это количество пирожков.

Количество пирожков	Цена за штуку
$n < 5$	100
$n \geq 5$ and $n < 10$	95
$n \geq 10$	90

Задание 3. Напишите функцию, которая принимает число n и в ответ отдает сумму чисел от 1 до n включительно. Например, $n = 5$, следовательно, функция должна вернуть $1 + 2 + 3 + 4 + 5 = 15$.

Тема 3. Строки

1.3. Работа со строками

- Операции со строками
 - Что такое строки и зачем они нужны
 - Литералы строк
 - Индексация и доступ к символам строки
 - Операции с использованием строк
 - Функции и методы строк
- Практическое задание.

Напишите программу, которая считает количество слов в тексте, введенном пользователем, и выведет на экран их количество.

Пример работы программы:

Введите текст: Написать программу, которая считает количество слов в тексте
Количество слов в тексте: 8

Тема 4. Списки

1.4. Работа со списками

- Операции со списками
 - Списки в Python
 - Списки упорядочены
 - Списки могут содержать произвольные объекты
 - Индексация списков
 - Вложенность списков
 - Методы списков
- Практическое задание

Напишите программу, которая принимает на вход список чисел и выводит на экран два списка: список четных чисел и список нечетных чисел из входного списка. Пример использования:

```
Введите числа через пробел: 1 2 3 4 5 6
Список четных чисел: [2, 4, 6]
Список нечетных чисел: [1, 3, 5]
```

Тема 5. Словари и коллекции

1.5. Работа со словарями и коллекциями

- Операции со словарем
 - Определение словаря
 - Доступ к значениям словаря
 - Индексация в словарях
 - Ограничения на ключи
 - Методы словарей
- Операции с коллекцией
 - Определение коллекции
 - set
 - dict
 - list
 - tuple
- Практическое задание.

Задание 1. Напишите программу, которая будет работать с информацией о книгах в библиотеке.

1. Создайте пустой словарь `library`.

2. Заполните словарь книгами, используя функцию `add_book(title, author, year, genre)`, которая принимает параметры: название книги (`title`), автор (`author`), год издания (`year`) и жанр (`genre`). Функция должна добавлять книгу в словарь `library` в виде пары «ключ — значение», где ключом будет название книги, а значением будет словарь с информацией о книге (автор, год издания и жанр).

3. Напишите функцию `print_books()`, которая будет выводить на экран все книги, которые есть в словаре `library` в формате «Название книги — Автор (Год издания, жанр)».

Результат выполнения программы:

```
Список книг в библиотеке:
Война и мир - Л. Толстой ( 1869 , Роман )
Преступление и наказание - Ф. Достоевский ( 1866 , Роман )
Мастер и Маргарита - М. Булгаков ( 1967 , Роман )
1984 - Дж. Оруэлл ( 1949 , Антиутопия )
```

Задание 2. Необходимо написать функцию, на вход которой поступает список целых чисел, и которая должна вернуть новый список, содержащий только четные целые числа, отсортированные в порядке возрастания.

Для примера список может быть таким:

```
lst = [3, 7, 12, 5, 14, 8, 9, 2]
```

На выходе функции список должен стать таким:

```
[2, 8, 12, 14]
```

Пример использования:

```
lst = [3, 7, 12, 5, 14, 8, 9, 2]
sort_evens(lst)
print(lst)
[2, 8, 12, 14]
```

Тема 6. Функции

1.6. Использование функций

- Использование функций
 - Определение функции
 - Важность функций
 - Вызовы функций и их определение
 - Передача аргументов в функцию
 - Возвращаемое значение
- Практическое задание.

Напишите программу-калькулятор, которая будет выполнять математические операции (+, -, *, /) в зависимости от выбора пользователя и выводить результат. Для реализации программы используйте функции. Пример работы программы:

```
Выберите операцию, которую хотите выполнить:
+ - сложение двух чисел
- - вычитание двух чисел
* - умножение двух чисел
/ - деление двух чисел
Введите знак операции: +
Введите первое число: 5
Введите второе число: 3.6
5.0 + 3.6 = 8.6
Выберите операцию, которую хотите выполнить:
+ - сложение двух чисел
- - вычитание двух чисел
* - умножение двух чисел
/ - деление двух чисел
Введите знак операции: /
Введите первое число: 6
Введите второе число: 0
На ноль делить нельзя!
```

Тема 7. Классы и объекты

1.7. Классы и объекты

- Классы и объекты как понятия
 - Все есть объект
 - Классы в Python
 - Объекты в Python
- Практическое задание
 - Измените стандартный объем стакана кофе на 250 мл. Добавьте возможность создать стакан сразу наполненным некоторым количеством кофе. Добавьте метод drink,

позволяющий выпить некоторое количество кофе из стакана. Не забудьте, что количество кофе в стакане не бесконечно. Пусть статус возвращает информацию в литрах.

Тема 8. Объектно-ориентированное программирование (ООП)

1.8. Принципы ООП

- Что такое объектно-ориентированное программирование (ООП)?
- Основопологающие принципы ООП:
 - абстракция
 - инкапсуляция
 - наследование
 - полиморфизм
- Практическое задание.
 - Опишите абстракцию «пользователь»: выберите несколько полей и создайте класс для получившейся модели.
 - Добавьте поле «Пароль» и методы для работы с ним: установить пароль, проверить пароль. Методы шифрования реализовывать не нужно, но помните про инкапсуляцию. Поле с паролем не должно быть публичным.
 - Добавьте поле «Баланс». Используя наследование, создайте классы «Покупатель» и «Продавец».
 - Создайте внешнюю функцию «Сделка» с параметром «Пользователь». При передаче в функцию баланс покупателя должен уменьшаться, а продавца увеличиваться. Попробуйте использовать полиморфизм для решения этой задачи.

Тема 9. Инструменты инкапсуляции в Python

1.9. Модификаторы доступа, свойства и дескрипторы

- Модификаторы доступа, свойства и дескрипторы
 - Модификаторы доступа
 - Дескрипторы
 - Свойства
 - Ключевые моменты
- Практическое задание
 - Создайте класс, моделирующий стакан кофе. Предусмотрите возможность нанести на стакан произвольную надпись. Длина текста не должна превышать 42 символов. Для валидации используйте дескриптор.
 - Создайте свойство для атрибута `_filled_ml`, определяющее наполненность стакана кофе. Не забудьте про валидацию.

Тема 10. Библиотеки

1.10. Использование библиотек

- Модули и пакеты
- Виртуальное окружение
- Библиотеки
- Практическое задание:
 1. Создайте пакет Python, добавьте в него два вложенных пакета с модулями внутри. Создайте в модуле одного из пакетов класс. Попробуйте его импортировать в модуле второго пакета с помощью абсолютного и относительного импорта:

```
top_level_package
package_one
module - здесь создаём класс
package_two
module - здесь импортируем
```
 2. Создайте два виртуальных окружения и установите в них `pydantic 1.9` и `1.10`.

3. Создайте классы Moscow, Ekaterinburg, Vladivostok. Свойство local_time должно возвращать текущее время в соответствующем городе. Воспользуйтесь библиотекой: <https://pypi.org/project/pytz/>

Тема 11. Исключения в Python

1.11. Обработка исключений

- Об исключениях
- Обработка исключений
- Дополнительные блоки
- Прокидывание исключений
- Практическое задание:

1. Создайте свой класс исключения ValidationError, наследуемый от ValueError.

Создайте его наследников: TooSmallValue и TooBigValue.

2. Создайте класс CupOfCoffee, атрибут size_ml и свойство filled_ml. Добавьте валидацию значения свойства filled_ml: оно не должно превышать значение атрибута size_ml и быть меньше 0. Используйте при валидации созданные ранее классы исключений.

3. Создайте внешнюю функцию, которая будет заполнять значение свойства filled_ml и обрабатывать исключения.

3.3 Календарный учебный график

Календарный график обучения является примерным, составляется и утверждается для каждой группы.

Срок освоения программы — 3 недели. Начало обучения — по мере набора группы. Примерный режим занятий: не более 8 академических часов в день, до 40 часов в неделю. Промежуточная и итоговые аттестации проводятся согласно графику.

Примерный график освоения программы:

№	Темы / дни	В Р	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
1.	Типы и структуры данных	РП	1																					
		СР		2																				
2.	Основные конструкции языка	РП			2																			
		СР				2	2																	
3.	Строки	РП						1																
		СР							2															
4.	Списки	РП								1														
		СР									2													
5.	Словари и коллекции	РП										1												
		СР											2											
6.	Функции	РП												1										
		СР														2								
7.	Классы и объекты	РП														1								
		СР															2							
8.	Объектно-ориентированное программирование	РП															1							
		СР																2						
9.	Инструменты инкапсуляции в Python	РП																	1					
		СР																		2				
10.	Библиотеки	РП																			1			
		СР																				2		
11.	Исключения в Python	РП																				1		
		СР																					2	
Итоговая аттестация		РП																						2

4. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ - ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

4.1. Требования к квалификации педагогических кадров, представителей предприятий и организаций, обеспечивающих реализацию образовательного процесса

Требования к образованию и обучению лица, занимающего должность преподавателя: высшее образование — специалитет или магистратура, направленность (профиль) которого, как правило, соответствует преподаваемому учебному курсу, дисциплине (модулю).

Дополнительное профессиональное образование — профессиональная переподготовка, направленность (профиль) которой соответствует преподаваемому учебному курсу, дисциплине (модулю).

Педагогические работники обязаны проходить в установленном законодательством Российской Федерации порядке обучение и проверку знаний и навыков в области охраны труда.

Рекомендуется обучение по дополнительным профессиональным программам по профилю педагогической деятельности не реже чем один раз в три года.

Требования к опыту практической работы: при несоответствии направленности (профиля) образования преподаваемому учебному курсу, дисциплине (модулю) — опыт работы в области профессиональной деятельности, осваиваемой слушателями или соответствующей преподаваемому учебному курсу, дисциплине (модулю).

Преподаватель: стаж работы в образовательной организации не менее одного года; при наличии ученой степени (звания) — без предъявления требований к стажу работы.

Особые условия допуска к работе: отсутствие ограничений на занятие педагогической деятельностью, установленных законодательством Российской Федерации

Прохождение обязательных предварительных (при поступлении на работу) и периодических медицинских осмотров (обследований), а также внеочередных медицинских осмотров (обследований) в порядке, установленном законодательством Российской Федерации

Прохождение в установленном законодательством Российской Федерации порядке аттестации на соответствие занимаемой должности.

4.2. Требования к материально-техническим условиям

Организация проводит занятия по адресу: г. Москва, ул. Суцевский Вал, д. 18. Аудитории для занятий расположены на 11-м этаже здания.

Все занимаемые помещения соответствуют обязательным нормам пожарной безопасности и требованиям санитарно-эпидемиологических служб. Помещения имеют централизованные системы водоснабжения, отопления и канализации. Воздухообмен помещений обеспечивается современными системами кондиционирования, за счет приточно-вытяжной вентиляционной системы.

Учебным центром СКБ Контур заключен договор с организацией общественного питания о возможности обеспечения слушателей питанием.

В учебной аудитории проводятся лекции и практические занятия. Аудитория оснащена столами и стульями, в составе учебного оснащения маркерная доска и флипчарт, в случае

необходимости подключается мультимедийный проектор, слушателям предоставляются компьютеры.

Компьютерная сеть учебного центра оснащена необходимым оборудованием для доступа в интернет по выделенному каналу. На каждом компьютере обеспечен постоянный доступ к компьютерной программе «Контур.Школа».

Для проведения вебинаров и онлайн-трансляций используется оснащенная современным оборудованием видеостудия:

- помещение оборудовано посадочными местами для спикера(ов);
- спикеру предоставляется персональный компьютер с соответствующими мультимедийными характеристиками (Intel Core i3 либо идентичные по характеристикам, оперативная память: от 4 Гб и выше для всех ОС), со стабильным соединением с сетью Интернет на скорости не менее 1 Мбит/с;
- видеочамера (максимальное разрешение видео — не менее 3840 x 2160).

Размещение материалов вебинаров и доступ к ним участников обеспечивает техническая платформа (сайт, система управления сайтом, другие технические средства):

1. Трансляция вебинара в режиме реального времени.
2. Хранение, систематизация записей вебинаров, с предоставлением участникам возможности просмотра записи онлайн.
3. Хранение, систематизация и доступ к скачиванию материалов учебных программ.
4. Напоминание участникам о предстоящем вебинаре за 1 час до начала мероприятия.
5. Использование защищенных соединений, передача и прием видео и звука по протоколам RTMP(S) или аналогичным.
6. Управление качеством и разрешением передаваемого/принимаемого видео вплоть до разрешения HD 720p на каждого участника мероприятия (адаптивный стриминг).
7. Обмен короткими текстовыми сообщениями (чат).
8. Осуществление записи мероприятий в формате, не требующем конвертации для проигрывания (mp4, AVI, WMA и т.д.).
9. Система регистрации на вебинар.
10. Техническое сопровождение проведения вебинара.
11. Отображение числа участников.
12. Техническая доступность услуги не менее 99,8% времени.
13. Устойчивость при проведении вебинара при одновременном подключении до 3000 участников.
14. Возможность участия пользователей на вебинарах в браузерах Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Apple Safari с установленным плагином Adobe Flash Player.
15. Передача аудио- и видеоинформации на персональные компьютеры участников реализована при скорости интернет-соединения не менее 134 кбит/с.

Основные функции программы Контур.Школа:

1. Размещение расписания и описания учебных программ и условий обучения.
2. Онлайн-трансляция учебных занятий с возможностью обратной связи.
3. Размещение тестов и проведение онлайн-тестирования.
4. Размещение и выбор образовательного контента и заданий для слушателей.

5. Хранение учебно-методических материалов.
6. Обратная связь слушателей к организаторам и преподавателям.
7. Автоматическая фиксация хода учебного процесса, промежуточных и итоговых результатов слушателей.
8. Хранение информации о ходе учебного процесса и результатов обучения в течение периода обучения.
9. Сбор и хранение заявок на обучение и сведений о слушателях.
10. Создание и актуализация контента и учебно-методических материалов.
11. Информационно-консультационное обслуживание слушателей.

4.3. Информационные и учебно-методические условия реализации дополнительной общеобразовательной программы - дополнительной общеразвивающей программы

Дополнительная общеобразовательная программа – дополнительная общеразвивающая программа обеспечивается учебно-методическими материалами по всем модулям образовательной программы.

Фонд учебно-научной библиотеки содержит основную и дополнительную учебную, учебно-методическую, научную литературу, справочно-библиографические и периодические издания (в том числе и на электронных носителях) по всем темам и дисциплинам реализуемой программы.

Нормативно-правовая база

1. Федеральный закон "Об информации, информационных технологиях и о защите информации" от 27.07.2006 №149-ФЗ
2. ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств» (утвержден приказом Росстандарта от 30 ноября 2010 г. N 631-ст);
3. ГОСТ Р ИСО/МЭК 14764-2002 «Информационная технология. Сопровождение программных средств» (принят постановлением Госстандарта РФ от 25 июня 2002 г. N 248-ст);
4. ГОСТ Р ИСО/МЭК 90003-2014 «Разработка программных продуктов. Руководящие указания по применению ИСО 9001:2008 при разработке программных продуктов» (утвержден приказом Росстандарта от 23 октября 2014 г. N 1405-ст).

Список литературы

1. Знакомство с Python. / [Бейдер Дэн]; Питер, 2023г. – 512 с.
2. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. / [Мэттиз Эрик]; Питер, 2022г. – 512 с.
3. Изучаем программирование на Python / [Бэрри Пол]; Эксмо, 2022г. – 624 с.
4. Программирование на Python в примерах и задачах. [Васильев А.Н.]; Бомбора, 2022 г. – 616 с.
5. Программирование на Python для начинающих перевод с английского. [МакГрат Майк]; Эксмо, 2023г. – 192 с.
6. Профессиональная разработка на Python / [Мэттью Уилкс]; ДМК Пресс, 2021 г. - 502 с.
7. Разработка требований к программному обеспечению. 3-е изд., дополненное / [Вигерс К., Битти Дж.]; БХВ; 2023 г. - 736 с.
8. Справочник PYTHON. Кратко, быстро, под рукой / [Дубовик Е. В., Кольцов Д. М.]; Наука и техника, 2021 г. - 288с.

Периодические издания

1. Журнал «Вестник компьютерных и информационных технологий», №2, 2023г.
<http://www.vkit.ru/index.php/archive-rus/1228-02-february>
2. Научно-практический журнал «Программные продукты и системы» №1, 2023г.
<http://www.swsys.ru/index.php>

Интернет-ресурсы

1. Основы Python. Научитесь думать как программист. Аллен Б. Дауни
https://library.samdu.uz/files/b0c333b5613b7c1710d62e0194dfc40e_%D0%94%D0%B0%D1%83%D0%BD%D0%B8%20%D0%90.%20-%20%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D1%8B%20Python%20-%202021.pdf
2. Учебник «Основы Python с нуля»
https://okpython.net/python/python_uchebnik/python_uchebnik.html?yclid=15535983162980302847

4.4. Общие требования к организации образовательного процесса.

При реализации дополнительной общеобразовательной программы – дополнительной общеразвивающей программы используется лекционно-семинарская система, практические занятия, индивидуальные и групповые консультации, мастер-классы и другие интерактивные формы обучения. Интегративно-дифференцированная организация занятий в процессе подготовки слушателей позволяет не только адаптировать образовательный процесс к индивидуальным особенностям и интересам слушателей, но и обеспечить свободу выбора и вариативность образования, сформировать у слушателей стремление к самообразованию, способствовать решению индивидуальных задач развития личности.

Особую значимость на уровне технологии обучения имеет организационно-педагогическое условие, предполагающее организацию самостоятельной работы слушателей как средство формирования профессиональных компетенций. Самостоятельная работа слушателей является обязательной составляющей образовательного процесса подготовки слушателей дополнительной общеобразовательной программы – дополнительной общеразвивающей программы.

Применение дистанционных образовательных технологий и электронного обучения

В учебном процессе могут использоваться следующие организационные формы учебной деятельности:

- обзорные (установочные) лекции с использованием дистанционных образовательных технологий;
- самостоятельная работа с СДО: работа с электронным учебником, просмотр видеолекций и др.;
- самостоятельная работа с программами контроля знаний (тестами);
- консультации (индивидуальные с применением электронных средств, групповые и предэкзаменационные);
- семинары с использованием дистанционных образовательных технологий;
- коллоквиумы;
- итоговые аттестационные работы.

5. СИСТЕМА ОЦЕНКИ КАЧЕСТВА ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

5.1. Формы контроля знаний и требования к его проведению

Для проведения промежуточной и итоговой аттестации программы разработан фонд оценочных средств по программе, являющийся неотъемлемой частью учебно-методического комплекса.

Объектами оценивания выступают:

- степень освоения теоретических знаний;
- уровень овладения практическими умениями и навыками по всем видам учебной работы, активность на занятиях.

Текущий контроль знаний слушателей проводится преподавателем, ведущим занятия в учебной группе, на протяжении всего обучения по программе.

Текущий контроль знаний включает в себя наблюдение преподавателя за учебной работой слушателей и проверку качества знаний, умений и навыков, которыми они овладели на определенном этапе обучения посредством выполнения упражнений на практических занятиях и в иных формах, установленных преподавателем.

Промежуточная аттестация — оценка качества усвоения слушателями содержания учебных блоков непосредственно по завершении их освоения, проводимая в форме зачета посредством тестирования или в иных формах, в соответствии с учебным планом и учебно-тематическим планом.

Итоговая аттестация — процедура, проводимая с целью установления уровня знаний, слушателей с учетом прогнозируемых результатов обучения и требований к результатам освоения образовательной программы. Итоговая аттестация слушателей осуществляется в форме зачета посредством тестирования.

Слушатель допускается к итоговой аттестации после изучения тем образовательной программы в объеме, предусмотренном для лекционных и практических занятий.

Слушателям, освоившим дополнительную общеобразовательную программу - дополнительную общеразвивающую программу по теме «Основы языка Python», и успешно прошедшим итоговую аттестацию, выдается свидетельство установленного образца с указанием названия программы, календарного периода обучения, длительности обучения в академических часах.

Для аттестации слушателей на соответствие их персональных достижений требованиям соответствующей ОП созданы фонды оценочных средств, включающие типовые задания, тесты и методы контроля, позволяющие оценить знания, умения и уровень приобретенных компетенций.

Фонды оценочных средств соответствуют целям и задачам программы подготовки специалиста, учебному плану и обеспечивают оценку качества общепрофессиональных и профессиональных компетенций, приобретаемых слушателями.

5.2. Критерии оценки знаний слушателей

Предмет оценивания (компетенции)	Объект оценивания (навыки)	Показатель оценки (знания, умения)
Основы написания программы	Использование языка Python для написания	Знания: <ul style="list-style-type: none">• Типы и структуры данных в языке Python.

на языке Python	программ	<ul style="list-style-type: none"> • Типы данных логических значений. • Основные конструкции языка: переменные, константы, оператор присваивания, условные операторы в языке Python. <p>Умения:</p> <ul style="list-style-type: none"> • Написать программы используя основные конструкции языка.
Работа с данными и объектами	Работа с данными и объектами с помощью языка программирования Python	<p>Знания:</p> <ul style="list-style-type: none"> • Особенности работы со строками, списками, словарями и коллекциями в Python. • Функции, классы и объекты. • Особенности применения инструментов инкапсуляции в Python. <p>Умения:</p> <ul style="list-style-type: none"> • Использовать словари и коллекции данных при разработке ПО. • Писать программы, которые используют функции. • Создавать и использовать собственные классы и методы, хранить состояние объектов в атрибутах. • Описывать публичные и внутренние интерфейсы. • Использовать свойства и дескрипторы для доступа к атрибутам объектов.
Использование библиотеки языка	Использование библиотек для повышения производительности кода	<p>Знания:</p> <ul style="list-style-type: none"> • Работа с модулями и пакетами в Python. • Виртуальное окружение. • Использование библиотек. <p>Умения:</p> <ul style="list-style-type: none"> • Импортировать модули и пакеты. • Создавать виртуальное окружение. • Импортировать, устанавливать и использовать библиотеки.
Исключения в Python	Создание кода с обработкой исключений	<p>Знания:</p> <ul style="list-style-type: none"> • Исключения в Python. • Обработка исключений. • Обработка разных классов исключений. • Прокидывание исключений во время обработки. <p>Умения:</p> <ul style="list-style-type: none"> • Обрабатывать исключения. • Использовать функциональность дополнительных блоков. • Прокидывать исключения во время обработки.

Критерии оценки аттестации – зачета:

1. Оценка «**Зачтено**» выставляется слушателю, продемонстрировавшему твердое и всесторонние знания материалы, умение применять полученные в рамках занятий практические навыки и умения. Достижения за период обучения и результаты текущей аттестации продемонстрировали отличный уровень знаний и умений слушателей. Не менее 80% правильных ответов при решении тестов.

2. Оценка «**Не зачтено**» выставляется слушателю, который в недостаточной мере овладел теоретическим материалом по дисциплине, допустил ряд грубых ошибок при выполнении практических заданий, а также не выполнил требований, предъявляемых к промежуточной аттестации. Достижения за период обучения и результаты текущей аттестации продемонстрировали неудовлетворительный уровень знаний и умений слушателя. Менее 80% правильных ответов при решении тестов.

5.3. Фонд оценочных средств

Тест к теме «Типы и структуры данных»

1. Какой тип данных в Python используется для хранения целых чисел?

1. float
2. str
3. int
4. bool

2. Какой тип данных в Python используется для хранения текстовой информации?

1. float
2. str
3. int
4. bool

3. Какой тип данных будет, если сложить целое 5 и дробное 3.14? (22171)

1. float
2. str
3. int
4. bool

4. Какая структура данных является неизменяемой?

1. Список (list)
2. Словарь (dict)
3. Кортеж (tuple)
4. Множество (set)

5. Какую структуру данных можно использовать для хранения элементов в стеке (LIFO - последний вошел, первый вышел)?

1. Список (list)
2. Кортеж (tuple)
3. Словарь (dict)
4. Множество (set)

Тест к теме «Основные конструкции языка»

2. Как объявить константу в Python?

1. var PI = 3.1415
2. const PI = 3.1415
3. final PI = 3.1415
4. PI = 3.1415

1) Что выведет следующий код?

```
x = 5
y = 10
```

```
if x > y:
    print("x больше чем y")
else:
    print("x меньше или равно y")
```

1. x больше чем y
2. x меньше чем y
3. x меньше или равно y
4. Ничего из вышеперечисленного

2) Какой будет равно z в результате следующего кода? (22176)

```
x = 10
y = 10
z = 0
if x > y:
    z = 5
elif x == y:
    z = 55
else:
    z = 555

print(z)
```

1. 0
2. 5
3. 55
4. 555

1. Какие типы циклов поддерживаются в Python?

1. for, while, do-while, repeat-until
2. for, while, until, foreach
3. for, while, do-while, continue
4. for, while

5. Каким оператором можно пропустить текущую итерацию в цикле в Python?

1. continue
2. next
3. skip
4. ignore

Тест к теме «Строки»

1. Как создать строку в Python?

1. string = "hello world"
2. string = 'hello world'
3. string = ""hello world""
4. все вышеперечисленные варианты верны

2. Как добавить строку к другой строке в Python?

1. str1 = "hello"
str2 = "world"
string = str1 + str2
2. str1 = "hello"
str2 = "world"
string = str1.append(str2)
3. str1 = "hello"

```
str2 = "world"
string = str1.expand(str2)
```

4. невозможно добавить строку к другой строке в Python

3. Как получить длину строки в Python?

1. `string = "hello world"`
`length = string.length()`
2. `string = "hello world"`
`length = len(string)`
3. `string = "hello world"`
`length = string.count()`
4. `string = "hello world"`
`length = string.size()`

4. Как проверить, содержится ли подстрока в строке?

1. `string = "hello world"`
`substring = "world"`
`if substring in string:`
 `print("Yes")`
`else:`
 `print("No")`
2. `string = "hello world"`
`substring = "planet"`
`if substring not in string:`
 `print("No")`
`else:`
 `print("Yes")`
3. `string = "hello world"`
`substring = "world"`
`if string.has(substring):`
 `print("Yes")`
`else:`
 `print("No")`
4. только а и б верны

5. Как заменить часть строки на другую строку в Python?

1. `string = "hello world"`
`new_string = string.replace("world", "planet")`
2. `string = "hello world"`
`new_string = string.swap("world", "planet")`
3. `string = "hello world"`
`new_string = string.switch("world", "planet")`
4. невозможно изменить строку в Python

<pre>str1 = "hello" str2 = "world" string = str1 + str2</pre>	1	<pre>str1 = "hello" str2 = "world" string = str1.append(str2)</pre>	2
<pre>str1 = "hello" str2 = "world" string = str1.expand(str2)</pre>	3	<pre>str1 = "hello" str2 = "world" string = str1.expand(str2)</pre>	4
невозможно добавить строку к другой строке в Python			5

Тест к теме «Списки»

1. Какой метод в Python используется для добавления элемента в конец списка?
 1. add()
 2. push()
 3. append()
 4. insert()
2. Какой метод в Python используется для удаления последнего элемента из списка?
 1. remove()
 2. delete()
 3. pop()
 4. discard()
3. Что из следующего верно для списков Python?
 1. Список может содержать объекты любого типа
 2. Все элементы в списке должны быть одного типа
 3. Эти списки одинаковые:

```
4. ['a', 'b', 'c']  
   ['c', 'a', 'b']
```
 5. Список может содержать одинаковые значения
4. Какой метод в Python используется для сортировки списка в порядке возрастания?
 1. sort()
 2. arrange()
 3. order()
 4. organize()
5. Какой метод в Python используется для создания нового списка на основе существующего?
 1. copy()
 2. clone()
 3. update()
 4. duplicate()

Тест к теме «Словари и коллекции»

1. Как получить список всех ключей в словаре?
 1. dict.get_keys()
 2. dict.keys()
 3. dict.all_keys()
 4. dict.key_list()
2. Что из следующего верно для словарей Python?
 1. Доступ к элементам осуществляется по их положению в словаре.
 2. Доступ к словарям осуществляется по ключу.
 3. Словарь может содержать любой тип объекта, кроме другого словаря.
 4. Словари изменяемы.
2. Какая структура данных Python является неизменяемой?
 1. tuple
 2. list
 3. set
 4. dict
3. Какой метод используется для объединения двух списков в Python?
 1. merge()
 2. join()

3. `append()`
4. `extend()`
4. Какая структура данных Python использует для хранения пары ключ-значение?
 1. `tuple`
 2. `list`
 3. `set`
 4. `dict`

Тест к теме «Функции»

1. Что такое функция в Python?
 1. Переменная со значением
 2. Блок кода, который можно вызвать из другого кода
 3. Ошибки в коде
 4. Программный модуль
2. Как объявить функцию в Python?
 1. `function myfunction():`
 2. `myfunction = function():`
 3. `def myfunction():`
 4. `declare myfunction as function():`
3. Что такое аргументы функции?
 1. Переменные, которые передаются в функцию при ее вызове
 2. Значения переменных, объявленных внутри функции
 3. Алгоритм работы функции
 4. Используемые модули
4. Как в Python можно вернуть значение из функции?
 1. функция возвращает значение неявно
 2. с помощью оператора `return`
 3. при помощи оператора `print`
 4. значение нельзя вернуть из функции
5. Как можно вызвать функцию в Python?
 1. Вызов функции осуществляется при помощи функции `myfunction()`
 2. Через ключевое слово `import`
 3. при помощи оператора `print`
 4. функция вызывается автоматически, как только программа запускается.

Тест к теме «Классы и объекты»

1. Что НЕ является объектом в Python?
 1. Целое число
 2. Функция
 3. Класс
 4. Все сущности в Python являются объектами
2. Что делает оператор `pass`?
 1. Ничего
 2. Кидает исключение
 3. Пропускает итерацию в цикле
 4. Запускает отладчик
3. С помощью какого ключевого слова создаются классы в Python?
 1. `cls`
 2. `class`
 3. `klass`
 4. `type`

6. Что означает одиночное нижнее подчеркивание в начале имени атрибута?
 1. Имя атрибута зарезервировано внутренними механизмами языка
 2. Ничего не означает
 3. Атрибут следует считать приватным
 4. Имена любых атрибутов нужно начинать с нижнего подчеркивания
8. Как создать атрибут экземпляра класса в Python?
 1. Объявить в виде поля класса
 2. Объявить переменную внутри конструктора
 3. Создать в конструкторе новое поле для self
 4. В Python есть только атрибуты самого класса

Тест к теме «Объектно-ориентированное программирование»

1. Что такое ООП?
 1. Подход к программированию, основанный на объектах и их взаимодействии
 2. Способ организации рабочих процессов
 3. Подход к программированию, при котором описывается ожидаемый результат, а не способ его достижения
 4. Язык программирования
2. Что из перечисленного НЕ является принципом ООП?
 1. Инкапсуляция
 2. Наследование
 3. Полиморфизм
 4. Все является
3. Что такое абстракция в ООП?
 1. Процесс разделения сложной задачи на простые
 2. Способ представления упрощенной модели объекта
 3. Детально проработанная точная модель объекта и всех его свойств
4. Что такое наследование в ООП?
 1. Процесс адаптации старого кода к новым условиям
 2. Способность абстракции перенимать свойства другой абстракции
 3. Процесс отмирания старого кода
5. Что такое инкапсуляция в ООП?
 - a. Способ объединения множества сущностей в одну
 - b. Изоляция внешнего представления от внутренней логики
 - c. Внедрение части кода на другом языке

Тест к теме «Инструменты инкапсуляции в Python»

1. Какие модификаторы доступа есть в Python?
 1. private
 2. protected
 3. private, protected и public
 4. В языке Python нет модификаторов доступа в их привычном понимании
2. С помощью какой переменной можно определить публичный интерфейс модуля?
 1. `__interface__`
 2. `__public__`
 3. `__all__`
 4. `__module__`
3. Для чего используется одиночное нижнее подчёркивание на конце имени?
 1. Не используется в Python
 2. Для обозначения внутреннего интерфейса
 3. Чтобы избежать конфликтов имён

4. Для переменных внутри функций и методов
4. Какие бывают дескрипторы?
 1. Только данных
 2. Данных и не данных
 3. Только не данных
 4. Ничего из перечисленного
7. Какие процессы доступа к атрибутам позволяет определить property?
 1. Чтение
 2. Запись
 3. Удаление
 4. Все перечисленные

Тест к теме «Библиотеки»

1. Какой файл используется при импорте пакета?
 1. `__init__.py`
 2. `__module__.py`
 3. `__package__.py`
 4. `__import__.py`
2. Какой из этих вариантов синтаксически правильный?
 1. `from hashlib import sha512`
 2. `from hashlib import sha512 as hash_func`
 3. `from hashlib import *`
 4. Все перечисленные
3. Относительно чего строится путь при относительном импорте?
 1. Корня проекта
 2. Рабочего каталога
 3. Текущего модуля
 4. Исполняемого файла
4. Разные виртуальные окружения позволяют использовать:
 1. Разные версии интерпретатора
 2. Разные версии библиотек
 3. Разные версии интерпретатора и библиотек
 4. Ничего из перечисленного
5. Какой из этих способов импорта не рекомендуется использовать?
 1. `import hashlib`
 2. `from hashlib import sha512`
 3. `from hashlib import sha512 as hash_func`
 4. `from hashlib import *`

Тест к теме «Исключения в Python»

1. От какого класса наследуются абсолютно все исключения в Python?
 1. `Error`
 2. `BaseError`
 3. `Exception`
 4. `BaseException`
2. Какой оператор /нужно использовать чтобы бросить исключение в Python?
 1. `throw`
 2. `raise`
 3. `fire`
 4. `try`
3. Как следует перехватывать исключения?

1. Указывая конкретные классы исключений
 2. Указывая класс Exception
 3. Не указывая никакие классы
 4. Любым из перечисленных способов
4. Как указать несколько классов в одном блоке except?
 1. С помощью вертикальной черты "|"
 2. С помощью оператора сложения "+"
 3. Поместить их в кортеж
 4. Нет такой возможности
 5. Когда вызывается блок else?
 1. Когда в блоке try было брошено исключение
 2. Когда исключение обработано в блоке except
 3. Когда в блоке try не было исключений
 4. Всегда

Итоговое тестирование

1. Какой тип данных в Python используется для хранения целых чисел?
 1. float
 2. str
 3. int
 4. bool
2. Какая структура данных является неизменяемой?
 1. Список (list)
 2. Словарь (dict)
 3. Кортеж (tuple)
 4. Множество (set)
3. Что выведет следующий код?

```
x = 5
y = 10
if x > y:
    print("x больше чем y")
else:
    print("x меньше или равно y")
```

1. x больше чем y
 2. x меньше чем y
 3. x меньше или равно y
 4. Ничего из вышеперечисленного
4. Каким оператором можно пропустить текущую итерацию в цикле в Python?
 1. continue
 2. next
 3. skip
 4. ignore
 5. Как добавить строку к другой строке в Python?
 1. str1 = "hello"
str2 = "world"
string = str1 + str2
 2. str1 = "hello"
str2 = "world"
string = str1.append(str2)
 3. str1 = "hello"

- ```
str2 = "world"
string = str1.expand(str2)
```
4. невозможно добавить строку к другой строке в Python
  6. Как заменить часть строки на другую строку в Python?
    1. `string = "hello world"`  
`new_string = string.replace("world", "planet")`
    2. `string = "hello world"`  
`new_string = string.swap("world", "planet")`
    3. `string = "hello world"`  
`new_string = string.switch("world", "planet")`
    4. невозможно изменить строку в Python
  7. Какой метод в Python используется для удаления последнего элемента из списка?
    1. `remove()`
    2. `delete()`
    3. `pop()`
    4. `discard()`
  8. Какой метод в Python используется для сортировки списка в порядке возрастания?
    1. `sort()`
    2. `arrange()`
    3. `order()`
    4. `organize()`
  9. Что из следующего верно для словарей Python?
    1. Доступ к элементам осуществляется по их положению в словаре.
    2. Доступ к словарям осуществляется по ключу.
    3. Словарь может содержать любой тип объекта, кроме другого словаря.
    4. Словари изменяемы.
  10. Какая структура данных Python использует для хранения пары ключ-значение?
    1. `tuple`
    2. `list`
    3. `set`
    4. `dict`
  11. Как в Python можно вернуть значение из функции?
    1. функция возвращает значение неявно
    2. с помощью оператора `return`
    3. при помощи оператора `print`
    4. значение нельзя вернуть из функции
  12. Как в Python можно вернуть значение из функции?
    1. функция возвращает значение неявно
    2. с помощью оператора `return`
    3. при помощи оператора `print`
    4. значение нельзя вернуть из функции
  13. Что НЕ является объектом в Python?
    1. Целое число
    2. Функция
    3. Класс
    4. Все сущности в Python являются объектами
  14. Что из перечисленного НЕ является принципом ООП?
    1. Инкапсуляция
    2. Наследование
    3. Полиморфизм

4. Все является
15. Что такое наследование в ООП?
  1. Процесс адаптации старого кода к новым условиям
  2. Способность абстракции перенимать свойства другой абстракции
  3. Процесс отмирания старого кода
16. С помощью какой переменной можно определить публичный интерфейс модуля?
  1. `__interface__`
  2. `__public__`
  3. `__all__`
  4. `__module__`
17. Какие процессы доступа к атрибутам позволяет определить `property`?
  1. Чтение
  2. Запись
  3. Удаление
  4. Все перечисленные
18. Разные виртуальные окружения позволяют использовать:
  1. Разные версии интерпретатора
  2. Разные версии библиотек
  3. Разные версии интерпретатора и библиотек
  4. Ничего из перечисленного
19. Как указать несколько классов в одном блоке `except`?
  1. С помощью вертикальной черты “|”
  2. С помощью оператора сложения “+”
  3. Поместить их в кортеж
  4. Нет такой возможности