

Муниципальное учреждение дополнительного образования  
«Центр внешкольной работы»

Рассмотрена на заседании  
методического совета МУДО ЦВР  
Протокол № 10 от 11.05.2023

Утверждаю  
директор МУДО ЦВР  
А.С. Девальд  
Приказ от 11.05.2023 № 253



**Дополнительная общеобразовательная общеразвивающая программа  
технической направленности  
«Python для начинающих»**

Возраст учащихся: 13-17 лет  
Срок реализации: 1 год

Автор-составитель:  
Полянская Ирина Ивановна,  
педагог дополнительного  
образования

г. Оленегорск  
2023 год

## Пояснительная записка

### Актуальность программы

По результатам исследования компании Jet Brains язык Python – самый изучаемый в 2019 году и один из самых востребованных на рынке труда. Python достаточно широко используется при изучении основ алгоритмизации и программирования в школьном курсе информатики. В частности, используется в качестве базового языка для изучения в УМК по информатике авторов Поляков К.Ю., Еремин Е.А., допущенном к использованию в общеобразовательных организациях Министерством просвещения РФ (Приказ Министерства Просвещения РФ от 28 декабря 2018 года № 345).

Изучение Python в школе откроет учащимся возможности дальнейшего развития в области IT и поможет профориентации в старших классах, пригодится в олимпиадах по программированию и решению заданий ЕГЭ.

Курс предполагает смешанный формат обучения. Сочетание групповой работы в учебном кабинете и индивидуальной работы в личном кабинете на онлайн-платформе позволяет учащимся выработать не только технические навыки программирования, но и навыки социального взаимодействия при работе над финальным проектом курса, а главное – научиться самостоятельно выстраивать свое профессиональное развитие.

Отличительные особенности дополнительной общеобразовательной общеразвивающей программы технической направленности программы «Python для начинающих» (далее – программа): авторский подход к организации образовательного процесса. Реализация программы предполагает проведение аудиторных занятий с использованием вариативных дистанционных ресурсов:

- авторских онлайн уроков, размещенных на платформе Stepik;
- авторских интерактивных заданий (опросов, викторин, дидактических игр и др.), размещенных на бесплатных специализированных сервисах Kahoot и Learning Apps;
- авторских проектных заданий для создания учащимися игр на языке Python.

Программа разработана в соответствии с основными нормативными документами:

- Федеральным законом от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;

- приказом Министерства просвещения Российской Федерации от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

- «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ» (письмо Министерства и науки Российской Федерации от 18.11.2015 № 09-3242);

- письмом Министерства образования и науки РФ от 25.07.2016 № 09-1790 «Рекомендации по совершенствованию дополнительных

образовательных программ, созданию детских технопарков, центров молодежного инновационного творчества и внедрению иных форм подготовки детей и молодежи по программам инженерной направленности»;

- Концепцией развития дополнительного образования детей до 2030 года (распоряжение Правительства Российской Федерации от 31.03.2022 г. № 678-р);

- «Стратегией развития воспитания в Российской Федерации на период до 2025 года» (распоряжение Правительства Российской Федерации от 29.05.2015 г. № 996-р);

- требованиями и нормами СанПиН 2.4.3648-20, 1.2.3685-21 и другими законодательными актами Российской Федерации.

**Направленность программы:** техническая.

**Цель** программы – формирование у учащихся навыков программирования на языке Python и самонаправленного обучения.

**Задачи:**

**Обучающие:**

- изучить основы программирования на языке Python;  
- научиться применять полученные знания для решения практических задач;

- научиться применять полученные знания для решения практических задач;

- повысить уровень самостоятельности в обучении (по четырехступенчатой шкале Г. Гроу).

**Развивающие:**

- развитие познавательного интереса к программированию;  
- развитие интеллектуальных способностей  
- формирование у учащихся инженерного и изобретательского мышления;

**Воспитательные:**

- воспитание личностных качеств: самостоятельности, уверенности в своих силах, креативности;

- формирование навыков межличностных отношений и навыков сотрудничества, навыков работы в группе, формирование культуры общения и ведения диалога;

- мотивация к выбору инженерных профессий, овладению технологическими компетенциями в различных областях фундаментальной науки и техники, создание установок инновационного поведения.

**Уровень программы:** стартовый.

**Возраст учащихся:** 13-17 лет.

**Форма реализации программы:** очная.

**Срок реализации программы:** 1 год.

**Объем программы** – 68 часов.

**Количество учащихся в группе:** 10-12 человек.

**Форма организации занятий** – групповая, при работе над проектами – групповая, парная.

**Режим занятий:** 1 раз в неделю по 2 академических часа.

**Виды учебных занятий и работ:** практические работы, беседы, лекции, тестирование.

**Ожидаемые результаты:**

**Предметные:**

В результате освоения программы учащиеся должны

**знать:**

- алгоритмические конструкции, лежащие в основе программирования;
- синтаксис операторов и их применение в решении задач;

**уметь:**

- программировать на языке Python;
- использовать инструменты разработки среды Wing;
- самостоятельно реализовывать проекты, связанные с разработкой игр.

Практическим результатом работы служит финальный проект каждого учащегося: 2D игра на движке Pygame, либо серия самостоятельно разработанных мини-проектов в консольном режиме языка Python.

**Метапредметные:**

- ставить учебные цели;
- формулировать достигнутый результат;
- планировать свою самостоятельную учебно-познавательную деятельность; выбирать индивидуальную траекторию достижения учебной цели;
- определять подходы и методы для достижения поставленной цели;
- отбирать необходимые средства для достижения поставленной цели;
- осуществлять самооценку промежуточных и итоговых результатов своей самостоятельной учебно-познавательной деятельности;
- проводить рефлексию своей учебно-познавательной деятельности.

**Личностные:**

- критическое отношение к информации и избирательность её восприятия;
- развитие любознательности, сообразительности при выполнении разнообразных заданий проблемного и эвристического характера;
- развитие внимательности, настойчивости, целеустремленности, умения преодолевать трудности;
- формирование профессионального самоопределения;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками.

Основные **формы организации занятий** программы «Python для начинающих»:

- практические занятия с использованием онлайн-платформы Stepik;

- работа в IDE «Wing 101»;
- прохождение опросов в приложениях Kahoot и Learning Apps;
- домашние практические занятия с использованием онлайн-платформы Stepik, направленные на отработку навыков программирования на языке Python.

### Методы обучения

- информационно-развивающий (самостоятельная дистанционная работа);
- проблемно-поисковый (проблемное изложение);
- репродуктивные (выполнение по образцу, по инструкции);
- творчески-воспроизводящие (вариативные упражнения, решение контекстов).

### Методическое обеспечение

- методические указания для педагога в текстовом виде,
- презентации с иллюстративным изложением теоретического материала;
- упражнения на платформе Stepik с теоретическим и практическим материалом языка Python;
- интерактивные проверочные задания в приложениях Kahoot и Learning Apps;
- подвижные игры, направленные на закрепление знаний, полученных на занятии;
- инструкции для проведения рефлексии процесса обучения с учащимися.

### Учебный план

№ п/п	Название раздела, темы	Количество часов			Формы аттестации/контроля
		всего	теория	практика	
1.	<b>Вводное занятие.</b> Знакомство с Python. Команды input() и print()	2	1	1	беседа, инструктаж, самостоятельная работа
2.	Параметры sep, end. Переменные. Комментарии. PEP 8	2	1	1	самостоятельная работа
3.	Работа с целыми числами	2	1	1	самостоятельная работа
4.	Условный оператор. Логические операции and, or, not	2	1	1	самостоятельная работа
5.	Вложенный и каскадный условный оператор	4	1	3	самостоятельная работа

6.	Типы данных int, float, str. Встроенные функции min(), max(), abs(). Оператор in	3	1	2	самостоятельная работа, зачет
7.	Цикл for. Функция range()	2	1	1	самостоятельная работа
8.	Частые сценарии при написании циклов. Расширенные операторы присваивания	3	1	2	самостоятельная работа
9.	Цикл с предусловием while	3	1	2	краткий опрос, самостоятельная работа
10.	Операторы break, continue, else	2	1	1	самостоятельная работа
11.	Вложенные циклы	5	1	4	инструктаж, самостоятельная работа
12.	Строковый тип данных: индексация и срезы	2	1	1	самостоятельная работа
13.	Методы строк	2	1	1	краткий опрос, самостоятельная работа
14.	Резервное время. Введение в списки	4	1	3	самостоятельная работа
15.	Основы работы со списками. Методы списков	4	1	3	краткий опрос, самостоятельная работа
16.	Вывод элементов списка. Строковые методы split() и join()	2	1	1	краткий опрос
17.	Методы списков. Списочные выражения	4	2	2	краткий опрос
18.	Функции	2	1	1	самостоятельная работа
19.	Локальные и глобальные переменные. Функции, возвращающие значения	4	1	3	краткий опрос
20.	Функции, возвращающие значения	4	1	3	краткий опрос
21.	Работа над проектом	8	1	7	краткий опрос, самостоятельная работа

22.	Заключительное занятие. Презентация проекта	2	-	2	анализ, подведение итогов
	<b>Итого:</b>	<b>68</b>	<b>22</b>	<b>46</b>	

### Содержание учебного плана

#### **Тема 1. Вводное занятие. Знакомство с Python. Команды input() и print()**

*Теория:* История языка Python. Сильные и слабые стороны Python. Python 2 VS Python 3.

Установка Python на компьютер. Установка Wing IDE на компьютер.

Вывод данных, команда print(). Ввод данных, команда input().

*Практика:* Самостоятельная работа, решение контекстов.

#### **Тема 2. Параметры sep, end. Переменные. Комментарии. PEP 8**

*Теория:* Необязательные параметры команды print(): sep, end. Переменные. Стандарт PEP 8. Комментарии. Решение задач.

*Практика:* Самостоятельная работа, решение контекстов.

#### **Тема 3. Работа с целыми числами**

*Теория:* Целочисленный тип данных. Преобразование строки к целому числу. Операции над целыми числами. Решение задач.

Операция возведения в степень. Операция нахождения остатка.

Операция целочисленного деления. Обработка цифр числа. Решение задач

*Практика:* Самостоятельная работа, решение контекстов.

#### **Тема 4. Условный оператор. Логические операции and, or, not**

*Теория:* Выбор из двух. Логические операции. Вложенные и каскадные условия.

*Практика:* Самостоятельная работа, решение контекстов.

#### **Тема 5. Вложенный и каскадный условный оператор**

*Теория:* Вложенный и каскадный условный оператор.

*Практика:* Самостоятельная работа, решение контекстов.

#### **Тема 6. Типы данных int, float, str. Встроенные функции min(), max(), abs(). Оператор in**

*Теория:* Числовые типы данных: int, float. Строковый тип данных. Модуль math

*Практика:* Самостоятельная работа, решение контекстов.

#### **Тема 7. Цикл for. Функция range()**

*Теория:* Цикл for. Цикл for: функция range.

*Практика:* Самостоятельная работа, решение контестов.

**Тема 8. Частые сценарии при написании циклов. Расширенные операторы присваивания**

*Теория:* Частые сценарии.

*Практика:* Самостоятельная работа, решение контестов.

**Тема 9. Цикл с предусловием while**

*Теория:* Цикл while. Цикл while: обработка цифр числа.

*Практика:* Самостоятельная работа, решение контестов.

**Тема 10. Операторы break, continue, else**

*Теория:* break, continue и else. Поиск ошибок и ревью кода.

*Практика:* Самостоятельная работа, решение контестов.

**Тема 11. Вложенные циклы**

*Теория:* Вложенные циклы.

*Практика:* Самостоятельная работа, решение контестов.

**Тема 12. Строковый тип данных: индексация и срезы**

*Теория:* Индексация. Срезы.

*Практика:* Самостоятельная работа, решение контестов.

**Тема 13. Методы строк**

*Теория:* Методы строк. Строки в памяти компьютера, таблица символов Unicode

*Практика:* Самостоятельная работа, решение контестов.

**Тема 14. Резервное время. Введение в списки**

*Теория:* Введение в списки. Основы работы со списками.

*Практика:* Самостоятельная работа, решение контестов.

**Тема 15. Основы работы со списками. Методы списков**

*Теория:* Методы списков. Часть 1.

*Практика:* Самостоятельная работа, решение контестов.

**Тема 16. Вывод элементов списка. Строковые методы split() и join()**

*Теория:* Вывод элементов списка. Методы строк: split, join

*Практика:* Самостоятельная работа, решение контестов.

**Тема 17. Методы списков. Списочные выражения**

*Теория:* Методы списков. Часть 2. Списочные выражения. Сортировка списков

*Практика:* Самостоятельная работа, решение контестов.



### **Тема 18. Функции**

*Теория:* Функции без параметров. Функции с параметрами.

*Практика:* Самостоятельная работа, решение констестов.

### **Тема 19. Локальные и глобальные переменные. Функции, возвращающие значения**

*Теория:* Локальные и глобальные переменные.

*Практика:* Самостоятельная работа, решение констестов.

### **Тема 20. Функции, возвращающие значения**

*Теория:* Функции с возвратом значения.

*Практика:* Самостоятельная работа, решение констестов.

### **Тема 21. Работа над проектом**

*Теория:* Модуль random. Числовая угадка. Магический шар 8. Генератор безопасных паролей. Шифр Цезаря. Калькулятор систем счисления. Угадка слов.

*Практика:* Создание проекта.

### **Тема 22. Заключительное занятие. Презентация проекта**

*Теория:* Подведение итогов.

*Практика:* Презентация проекта.

### **Материально-техническое обеспечение**

- 10-15 рабочих мест: компьютеры на каждое рабочее место;
- проектор, аудио колонки;
- Интернет-соединение, скорость загрузки не менее 2 Мбит/сек;
- меловая, магнитно-маркерная доска или флипчарт;
- общие условия в соответствии с СанПиН 2.4.4.3172-14.

### **Требования к ПО:**

- операционная система Windows 7 или моложе / MacOS / Unix-based системы с поддержкой протокола HTML5;
- приложения Google Chrome, Gimp, Brackets;
- интерактивная оболочка (бесплатная IDE Wing101 или аналог)

### **Формы аттестации и оценочные материалы**

Аттестация проводится в форме выполнения индивидуальных и групповых заданий по пройденному материалу. Контроль в указанной форме осуществляется как промежуточный, так и итоговый.

Для заданий на онлайн-тренажере указан необходимый минимум -70% выполненных заданий, чтобы тема считалась пройденной успешно, и был открыт доступ к следующей теме.

После каждой темы в онлайн-курсе стоит итоговая работа: от учащегося требуется в ограниченное время (три часа) решить набор задач по пройденной теме. В среднем, учащийся справляется с решением за 30 минут. В конце курса, по итогам работы над групповыми и индивидуальными проектами проводится обсуждение результатов в коллективе с опорой на чек-лист, исправление ошибок и, тем самым, коррекция и закрепление полученных знаний.

### **Список литературы для педагога и учащихся**

1. Васильев А.Н. Python на примерах: практ. курс /А.Н. Васильев - Наука и Техника, 2019 - 432 с.
2. Прохоренок Н.А. Python 3: самое необходимое: практ. курс /Н.А. Прохоренок В.А. Дронов. - БХВ-Петербург, 2019 - 608 с.
3. Гэддис Т. Начинаем программировать на Python: учебник/Т. Гэддис. - БХВ-Петербург, 2019 - 768 с.
4. Седжвик Р. Программирование на языке Python /Р. Седжвик, К. Уэйн, Р. Дондеро. - Вильямс, 2017 - 736 с.
5. Харрисон М. Как устроен Python: практ. курс / М. Харрисон. - Питер, 2002 - 272 с.

### **Электронные ресурсы**

1. Курс [Поколение Python: курс для начинающих](https://stepik.org/course/58852/syllabus) на платформе Stepik. [Электронный ресурс] // Режим доступа <https://stepik.org/course/58852/syllabus>. (Дата обращения: 15.04.2022 г.).

Программу составил  
педагог дополнительного образования  
Центра внешкольной работы

И.И. Полянская

Календарный учебный график

к дополнительной общеобразовательной общеразвивающей программе «Python для начинающих» (стартовый уровень освоения)

№ п/п	Месяц	Число	Время проведения	Форма занятия	Количество часов	Тема занятия	Место проведения	Форма контроля
1.	сентябрь		по расписанию	беседа, практическое занятие	2	Знакомство с Python. Команды input() и print(). Решение констестов	Квантолаб Строительная, 65	беседа, инструктаж, самостоятельная работа
2.	сентябрь		по расписанию	беседа, практическое занятие	2	Параметры sep, end. Переменные. Комментарии. Решение констестов	Квантолаб Строительная, 65	самостоятельная работа
3.	сентябрь		по расписанию	беседа, практическое занятие	2	Работа с целыми числами. Решение констестов	Квантолаб Строительная, 65	самостоятельная работа
4.	октябрь		по расписанию	беседа, практическое занятие	2	Условный оператор. Логические операции and, or, not. Решение констестов	Квантолаб Строительная, 65	самостоятельная работа
5.	октябрь		по расписанию	беседа, практическое занятие	2	Вложенный и каскадный условный оператор. Решение констестов	Квантолаб Строительная, 65	самостоятельная работа
6.	октябрь		по расписанию	беседа, практическое занятие	2	Вложенный и каскадный условный оператор. Решение констестов	Квантолаб Строительная, 65	самостоятельная работа
7.	октябрь		по расписанию	беседа, практическое занятие	2	Типы данных int, float, str. Встроенные функции min(), max(), abs(). Оператор in. Решение констестов	Квантолаб Строительная, 65	самостоятельная работа
8.	ноябрь		по расписанию	беседа, практическое занятие	2	Типы данных int, float, str. Встроенные функции min(), max(), abs(). Оператор in. Цикл for. Функция range(). Решение констестов		зачёт, самостоятельная работа
9.	ноябрь		по расписанию	беседа, практическое занятие	2	Цикл for. Функция range(). Частые сценарии при написании циклов. Расширенные операторы присваивания. Решение констестов	Квантолаб Строительная, 65	самостоятельная работа

10.	ноябрь		по расписанию	беседа, практическое занятие	2	Частые сценарии при написании циклов. Расширенные операторы присваивания. Решение контестов	Квантолаб Строительная, 65	самостоятельная работа
11.	ноябрь		по расписанию	беседа, практическое занятие	2	Цикл с предусловием while. Решение контестов	Квантолаб Строительная, 65	краткий опрос
12.	ноябрь		по расписанию	практическое занятие	2	Цикл с предусловием while Операторы break, continue, else	Квантолаб Строительная, 65	самостоятельная работа
13.	декабрь		по расписанию	беседа, практическое занятие	2	Операторы break, continue, else Вложенные циклы. Решение контестов	Квантолаб Строительная, 65	краткий опрос, самостоятельная работа, инструктаж,
14.	декабрь		по расписанию	практическое занятие	2	Вложенные циклы	Квантолаб Строительная, 65	самостоятельная работа
15.	декабрь		по расписанию	практическое занятие	2	Вложенные циклы	Квантолаб Строительная, 65	самостоятельная работа
16.	декабрь		по расписанию	беседа, практическое занятие	2	Строковый тип данных: индексация и срезы. Решение контестов	Квантолаб Строительная, 65	самостоятельная работа
17.	январь		по расписанию	беседа, практическое занятие	2	Методы строк. Решение контестов	Квантолаб Строительная, 65	краткий опрос, самостоятельная работа
18.	январь		по расписанию	беседа, практическое занятие	2	Резервное время. Введение в списки. Решение контестов	Квантолаб Строительная, 65	самостоятельная работа
19.	январь		по расписанию	практическое занятие	2	Резервное время. Введение в списки	Квантолаб Строительная, 65	самостоятельная работа
20.	февраль		по расписанию	беседа, практическое занятие	2	Основы работы со списками. Методы списков. Решение контестов	Квантолаб Строительная, 65	краткий опрос, самостоятельная работа
21.	февраль		по расписанию	практическое занятие	2	Основы работы со списками. Методы списков	Квантолаб Строительная, 65	самостоятельная работа

22.	февраль		по расписанию	беседа, практическое занятие	2	Вывод элементов списка. Строковые методы split() и join(). Решение контестов	Квантолаб Строительная, 65	краткий опрос
23.	февраль		по расписанию	беседа, практическое занятие	2	Методы списков. Списочные выражения. Решение контестов	Квантолаб Строительная, 65	краткий опрос
24.	март		по расписанию	беседа, практическое занятие	2	Методы списков. Списочные выражения. Решение контестов	Квантолаб Строительная, 65	краткий опрос
25.	март		по расписанию	беседа, практическое занятие	2	Функции. Решение контестов	Квантолаб Строительная, 65	самостоятельная работа
26.	март		по расписанию	беседа, практическое занятие	2	Локальные и глобальные переменные. Функции, возвращающие значения. Решение контестов	Квантолаб Строительная, 65	краткий опрос
27.	март		по расписанию	практическое занятие	2	Локальные и глобальные переменные. Функции, возвращающие значения	Квантолаб Строительная, 65	краткий опрос
28.	апрель		по расписанию	беседа, практическое занятие	2	Функции, возвращающие значения. Решение контестов	Квантолаб Строительная, 65	краткий опрос
29.	апрель		по расписанию	практическое занятие	2	Функции, возвращающие значения	Квантолаб Строительная, 65	краткий опрос
30.	апрель		по расписанию	беседа, практическое занятие	2	Работа над проектом. Решение контестов	Квантолаб Строительная, 65	краткий опрос
31.	апрель		по расписанию	беседа, практическое занятие	2	Работа над проектом. Решение контестов	Квантолаб Строительная, 65	самостоятельная работа
32.	май		по расписанию	беседа, практическое занятие	2	Работа над проектом. Решение контестов	Квантолаб Строительная, 65	краткий опрос, самостоятельная работа
33.	май		по расписанию	практическое занятие	2	Работа над проектом	Квантолаб Строительная, 65	краткий опрос

34.	май		по расписанию	итоговое занятие	2	Заключительное занятие. Презентация проекта	Квантолаб Строительная, 65	анализ, подведение итогов
<b>Итого:</b>					<b>68</b>			

## Оценочные материалы Тестирование

### 1. Каким ключевым словом обозначают функцию?

- с помощью инструкции f
- с помощью инструкции def
- с помощью инструкции function
- с помощью инструкции func

### 2. Как создаются анонимные функции?

- с помощью инструкции def
- с помощью инструкции anonim
- с помощью инструкции functionanonimus
- с помощью инструкции lambda

### 3. Что означает оператор or?

- логическое ИЛИ
- выполняет проверку условия
- логическое И
- логическое Равно

### 4. Какие аргументы необходимо передать функции average?

```
def average (x, y, z, q) :  
    sum =  
    return sum/4 x+y+z+q
```

- x,y,z,q
- x.y
- z,q
- sum

### 5. В каком виде представлены данные в python?

- в виде функций
- в виде объектов
- в виде массивов
- в двоичном виде

### 6. переменная name- это

- name- не переменная
- одинаковые переменные
- равные переменные
- Name- не переменная

**7. Какой конструкцией обрабатываются исключения?**

1. ifelse
2. try- except
3. def
4. for

**8. Как осуществить поиск в строке?**

1. командой Select
2. методом search
3. методом find()
4. с помощью orderby()

**9. Как запустить python- скрипт с параметрами?**

1. параметры передать нельзя
2. просто написать параметры в программе - будет работать
3. с помощью модуля os
4. с помощью модуля sys

**10. Какая конструкция открывает файл?**

1. read(path, 'r', encoding='UTF-8')
2. file(path, 'r', encoding='UTF-8')
3. open(path, 'r', encoding='UTF-8')

**11. Что такое атрибут?**

1. это число классов
2. это класс
3. это переменная класса

**12. Как передать данные функции?**

1. данные передаются в других функциях
2. данные передаются функции при её вызове в скобках
3. данные передаются через класс
4. данные в python нельзя передать в функции

**13. Как безопаснее работать с файлом?**

1. оба варианта безопасны
2. withopen(...) as f
3. f = open(...)

**14. Как добавить элемент в список?**



1. с помощью метода `.append()`
2. в список нельзя добавлять новые элементы
3. с помощью метода `.add()`

**15. Как в регулярных выражениях проверить соответствие строки “string” регулярному выражению “pattern”?**

1. `find`
2. `orderby`
3. `listen`
4. `match (pattern, string)`

**16. Какая команда печатает данные пользователя?**

1. `input()`
2. `print()`
3. `while()`
4. `foreach()`

**17. Как обойти всю последовательность?**

1. с помощью `forin`
2. с помощью `foreach`
3. с помощью `while`

**18. Что такое класс?**

1. это массив с атрибутами
2. это пользовательский тип, состоящий из методов и атрибутов
3. это список с атрибутами
4. в pythonнет классов

**19. Что такое модуль?**

1. отдельная функция
2. кусок кода
3. функционально законченный фрагмент программы, оформленный в виде отдельного файла с исходным кодом или поименованной непрерывной её части, предназначенный для использования в других программах

**20. Что такое кортеж?**

1. массив списков
2. неизменяемый список
3. изменяемый список

**21. Где в функции задано значение по умолчанию?**

1. `defwelcome(name="Инкогнито")`

2. `def welcome(name: "Инкогнито")`
3. `def (Инкогнито)`
4. `def (паше, Инкогнито)`

## 22. Является ли итератором данный класс?

```
class IterObj:
    def __init__(self, start=0):
        self.i = start
    def __next__(self):
        self.i += 1
        if self.i <= 5:
            return self.i
        else:
            raise StopIteration
```

1. нет
2. да
3. в python нет итераторов

## 23. Что выведет следующий код?

```
a = 20
b = a + 10
print(b)
```

1. 10
2. 5
3. 30
4. 20

## 24. Как получить элемент из списка?

1. по ключу
2. с помощью метода `.get()`
3. по индексу

## 25. Что делает метод словаря `popitem()`?

1. выводит весь словарь
2. возвращает первый элемент словаря
3. выводит случайный элемент словаря
4. удаляет элемент и возвращает пару (ключ, значение)

**26. В чем отличие глобальных и локальных переменных?**

1. нет разницы
2. локальные переменные выполняются во всем классе, глобальные только в функции
3. локальные переменные существуют только во время выполнения функции. Глобальные переменные - переменные, объявленные в основной части программы, т.е. вне функции

**27. Что выполнит данный код после импорта random?**

```
lst = []  
for _ in range(10):  
    lst.append(random.randint(-10, 10))
```

1. Вернет ошибку
2. Создаст пустой строковый массив и инициализирует его
3. Сгенерирует список из 10 случайных чисел
4. Сгенерирует список из 10 случайных чисел в диапазоне от -10 до 9

**28. Что такое итераторы?**

1. в pythonнет итераторов
2. итераторы - это специальные объекты, представляющие последовательный доступ к данным изконтейнера
3. специальная функция, означающая какие функции выполнить нельзя
4. итераторы - это специальные объекты, представляющие доступ к данным из контейнера по данным из функции

**29. Что делает функция lerp?**

1. преобразует строку в нижний регистр
2. очищает строку
3. преобразует строку в верхний регистр
4. она принимает любую последовательность в качестве аргумента и возвращает её длину

**30. Интерпретируемый язык программирования это**

1. язык программирования, в котором есть технология увеличения производительности программных систем, использующих байт-код, путём компиляции байт-кода в машинный код или в другой формат непосредственно во время работы программы
2. язык программирования, который преобразует эквивалентную программу на низкоуровневом языке, близком машинному коду
3. язык программирования, в котором исходный код программы не преобразуется в машинный код для непосредственного выполнения центральным процессором, а исполняется с помощью специальной программы-интерпретатора