

Министерство образования Свердловской области
Государственное автономное профессиональное образовательное учреждение
Свердловской области «Екатеринбургский монтажный колледж»

УТВЕРЖДАЮ:

Директор

В.Н.Чистяков

« 17 »

апреля

2025г.



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
ОТКРОЙ МИР ПРОГРАММИРОВАНИЯ С НУЛЯ НА PYTHON**

Направленность программы: Техническая

Категория слушателей: школьники, студенты СПО

Объем: 72 академ. часа

Срок: 6 месяцев

Форма обучения: Очная

Организация процесса обучения: с применением ДОТ

Екатеринбург, 2025

Актуальность программы «Открой мир программирования с нуля на Python » обусловлена чрезвычайной практической значимостью изучения технологий написания программ. Специалисты по языку программирования Python одни из самых востребованных на рынке труда. В то же время Python легок в освоении, что позволяет рекомендовать данный курс самому широкому кругу обучающихся, в том числе и школьникам. Курс построен таким образом, чтобы обучающийся смог сформировать базис знаний и применять конструкции языка программирования Python для решения широкого круга задач. Программа направлена на формирование навыков разработки программных решений, изучение основ программирования в Питоне , а также на знакомство с возможностями Питона по функциональному программированию.

Цель курса - получение практических навыков, требующихся для успешного понимания концепций программирования Обучающиеся получают базовые знания по решению задач с помощью Python. Научатся разрабатывать консольные приложения, освоят работу со встроенными и внешними библиотеками. Результатом обучения будет получение слушателями следующих базовых знаний:

- умение использовать термины программирования;
- знание синтаксиса языка Python;
- знание основных алгоритмических конструкций и структур языка;
- умение создавать программы для решения алгоритмических задач;
- получение навыков выбора способа представления данных в зависимости от поставленной задачи

В результате освоения и успешного завершения данного курса обучающийся будет готов к самостоятельному изучению других языков программирования и созданию собственных приложений.

Разработчик(и): Грибова Анна Владимировна Преподаватель

Организация: Государственное автономное профессиональное образовательное учреждение Свердловской области "Екатеринбургский монтажный колледж"

Оглавление

1. Общая характеристика программы	4
1.1. Нормативно-правовые основания разработки программы	4
1.2. Цели реализации программы	4
1.3. Требования к слушателям	4
1.4. Требования к результатам освоения программы	4
1.5. Форма документа.....	4
2. Учебный план	5
3. Календарный учебный график.....	6
4. Программы учебных модулей.....	7
5. Организационно-педагогические условия реализации программы.....	8
5.1. Материально-техническое обеспечение	8
5.2. Кадровое обеспечение	8
5.3. Организация образовательного процесса	8
5.4. Информационное обеспечение обучения	8
6. Контроль и оценка результатов освоения программы	9
Фонд оценочных средств.....	10

1. Общая характеристика программы

1.1. Нормативно-правовые основания разработки программы

Программа разработана в соответствии с:

- Федеральным законом от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- Приказом Министерства образования и науки Российской Федерации от 20.01.2014 № 22 «Об утверждении перечней профессий и специальностей среднего профессионального образования, реализация образовательных программ по которым не допускается с применением исключительно электронного обучения, дистанционных образовательных технологий»;
- Приказом Министерства образования и науки Российской Федерации (Минобрнауки России) от 1 июля 2013 г. N 499 г. Москва "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам"
- Приказом Министерства образования и науки Российской Федерации от 14.07.2013 № 464 «Об утверждении порядка организации и осуществления образовательной деятельности по образовательным программам среднего профессионального образования»;
- Письмом от 24 июня 2014г. № АК-1666/05 "Об установлении соответствий при утверждении новых перечней профессий, специальностей и направлений подготовки указанным в предыдущих перечнях профессий, специальностей и направлений подготовки"; спецификация стандартов Ворлдскиллс по компетенции «Программные решения для бизнеса»;
- профессиональный стандарт 06.001 "Программист» (утвержден приказом Минтруда России от 18.12.2013 года, регистрационный N 30635);

1.2. Цели реализации программы

Целью программы является приобретение профессионального опыта разработки приложений на Python. Результатом освоения курса является создание программных решений. В процессе обучения слушатель осваивает следующие виды работ: составление алгоритмов и запись их на языке программирования Python, разработка программных модулей в соответствии с техническим заданием

1.3. Требования к слушателям

К освоению программы допускаются школьники и лица, получающие среднее профессиональное образование. Медицинские ограничения регламентированы перечнем медицинских противопоказаний Минздрава России.

1.4. Требования к результатам освоения программы

Результатом освоения программы является освоение следующих компетенций:

Код	Наименование профессиональной компетенции
ПК 1.1	Составление алгоритмов и запись их на языке программирования Python
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием

1.5. Форма документа

По результатам освоения программы выдается: Сертификат

2. Учебный план

Наименование компонентов программы	Объем программы (академические часы)							
	Всего	В том числе с применением ДОТ и ЭО	Самостоятельная работа	Консультация	Нагрузка во взаимодействии с преподавателями			
					Теоретическое обучение	Практические и лабораторные работы	Практика (стажировка)	Промежуточная аттестация, форма
Модуль 1 Структуры данных Python	71					70		1, Зачёт с оценкой
Итоговая аттестация	1							Зачет с оценкой
Итого по программе	72							

3. Календарный учебный график

Компоненты программы	Вид учебной нагрузки	Временные параметры (месяцев)																								Всего
		1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	
Модуль 1 Структуры данных Python	Аудиторное обучение	2	2	2	2	4	4	4	4	4	4	4		4	4	4	4	4	4	4	4	2	2	2		70
	Промежуточная аттестация																							1		1
Итоговая аттестация	Зачет с оценкой																							1		1
Итого в неделю		2	2	2	2	4	4	4	4	4	4	4		4	4	4	4	4	4	4	4	2	2	2	2	72

4. Программы учебных модулей

4.1. Модуль 1. Структуры данных Python

Модуль направлен на изучение основ языка Python, составление алгоритмов и реализацию их на языке Python. Слушатели познакомятся с конструкциями языка программирования Python для решения широкого круга задач. Программа модуля направлена на формирование навыков разработки программных решений, изучение способов реализации базовых алгоритмов в Питоне, а также на знакомство с возможностями Питона по функциональному программированию. Слушатели изучат следующие темы:

1. Типы данных;
2. Условные конструкции;
3. Циклические конструкции;
4. Списки;
5. Строки;
6. Словари;
7. Функции;
8. Работа с текстовыми файлами;
9. Работа с Json

Модуль направлен на знакомство слушателей с основными В программе предусмотрены практические занятия. Модуль завершается итоговым тестом, который позволит продемонстрировать освоение основных алгоритмических конструкций языка Python, основных структур, используемых при разработке программы.

4.1.1. Цели реализации модуля

Целью модуля является формирование базовых знаний, умений и навыков решения наиболее важных и часто встречаемых на практике задач по

программированию на языке программирования Python.

Модуль направлен на :

1. изучение основ языка Python: ввод-вывод данных, условные операторы, циклы, структуры данных, функции;
2. получение базовых знаний по решению задач с помощью Python;
3. освоение работы со встроенными и внешними библиотеками;
4. составление алгоритмов и запись их на языке Python;
5. решение практических алгоритмических задач.

4.1.2. Требования к результатам освоения модуля

Результатом освоения модуля является освоение следующих компетенций:

Код	Наименование профессиональной компетенции
ПК 1.1	Составление алгоритмов и запись их на языке программирования Python
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием

В результате освоения модуля слушатель должен:

- иметь практический опыт:

В результате освоения модуля слушатель должен иметь практический опыт:

- по работе с операторами ввода-вывод данных;
- по работе с условными операторами;
- по работе с циклами;
- по работе со структурами данных;
- по работе с функциями;
- по работе с текстовыми файлами;

- по работе с json;
- по реализации базовых алгоритмов на примере языка Python

- знать:

- базовые алгоритмические конструкции языка Python;
- базовые алгоритмы управления и запись их на языке Python;
- правила оформления программ на языке Python;
- основные этапы разработки программного обеспечения;

- уметь:

- составлять алгоритмы управления и выполнять запись их на языке Python.
- применять основные алгоритмические конструкции языка для решения прикладных задач;
- формировать алгоритмы разработки программных модулей в соответствии с заданием;
- создавать программу по разработанному алгоритму;
- выполнять отладку программы;
- выполнять оптимизацию и рефакторинг программного кода;
- работать со встроенными и внешними библиотеками.

4.1.3. Программа модуля

Наименование разделов, тем модуля	Содержание обучения по темам, наименование и тематика практических занятий, самостоятельной работы. Вид учебных занятий. Виды выполняемых работ.	Объем часов
1	2	3

Тема "Основные операторы языка и скалярные типы данны"	Содержание: Изучение типов: int, float, bool, str. Изучение основных алгоритмических конструкций	22
	<i>Практическое занятие</i> Изучение основ языка программирования Python. Блоки и отступы при написании кода программы. Операторы ввода/вывода. Типы данных. Использование f-строк. Понятие переменной. Оператор присваивания. Математические операции. Встроенные математические функции, подключение библиотеки math.	2
	<i>Практическое занятие</i> Изучение основ языка программирования Python. Совместимость и преобразование типов. Типы данных: целые числа, дробные числа, строки, логический тип данных. Генератор случайных чисел. Операции с целыми числами. Операции с вещественными числами. Логический тип данных. Проверка типа введенных данных.	2
	<i>Практическое занятие</i> Создание линейных программ на Python. Обработка исключительных ситуаций программы. Конструкция try-except-finally-else.	2
	<i>Практическое занятие</i> Знакомство с разветвляющимися конструкциями. Логические переменные. Логические выражения и операторы. Операции сравнения. Условные конструкции языка: неполная форма, полная форма, вложенные операторы.	2
	<i>Практическое занятие</i> Работа с разветвляющимися конструкциями. Составные условия. Каскадные проверки условий	2
	<i>Практическое занятие</i> Бесконечный цикл. Досрочное прерывание цикла. Break и continue. Программирование с использованием операторов цикла.	2

Тема "Структурированные типы данных языка Python"	<i>Практическое занятие</i> Циклические конструкции Python (Цикл While). Программирование с использованием операторов цикла.	2
	<i>Практическое занятие</i> Циклические конструкции Python (цикл For). Программирование с использованием операторов цикла.	2
	<i>Практическое занятие</i> Циклические конструкции Python (цикл For). Проверка условий в циклах. Программирование с использованием операторов цикла.	2
	<i>Практическое занятие</i> Знакомство с понятием "функция" . Применение функций. Создание процедур и функций. Вызов подпрограммы из тела основной программы. Формальные и фактические параметры. Локальные и глобальные переменные. Возврат значений из функции.	2
	<i>Практическое занятие</i> Создание собственных функций (оператор def). Решение прикладных программ	2
	Содержание: Знакомство со структурированными типами: строки, списки, словари, файлы, множества	48
	<i>Практическое занятие</i> Циклический перебор символов строки. Знакомство с понятиями "индекс" и "срез"	2
	<i>Практическое занятие</i> Применение методов и функций для работы со строками	2
	<i>Практическое занятие</i> Применение методов и функций для работы со строками	2

<i>Практическое занятие</i>	
Работа со списками в Python. Методы и функции работы со списками	2
<i>Практическое занятие</i>	
Работа со списками в Python. Использование цикла for при работе со списками. Изменение элементов списка	2
<i>Практическое занятие</i>	
Работа со списками в Python. Решение прикладных задач	2
<i>Практическое занятие</i>	
Работа со списками в Python. Решение прикладных задач	2
<i>Практическое занятие</i>	
Работа со списками в Python. Решение прикладных задач	2
<i>Практическое занятие</i>	
Использование регулярных выражений при работе со строками , списками	2
<i>Практическое занятие</i>	
Библиотека random при работе со списками в Python	2
<i>Практическое занятие</i>	
Работа с текстовыми файлами в Python. Знакомство с популярными режимами работы с текстовыми файлами.	2
<i>Практическое занятие</i>	
Работа с текстовыми файлами в Python. Чтение информации из файла. Формирование списка из данных текстового файла. Запись информации в файл	2
<i>Практическое занятие</i>	
Работа с текстовыми файлами в Python. Решение	2

прикладных задач.	
<i>Практическое занятие</i> Работа с текстовыми файлами в Python. Решение прикладных задач.	2
<i>Практическое занятие</i> Работа с текстовыми файлами в Python. Решение прикладных задач.	2
<i>Практическое занятие</i> Работа с текстовыми файлами в Python. Решение прикладных задач.	2
<i>Практическое занятие</i> Работа с текстовыми файлами в Python. Решение прикладных задач.	2
<i>Практическое занятие</i> Знакомство с типом данных "словарь" в Python. Отличия словаря и от списка. Понятия "ключ" и "значение". Получение доступа к элементу словаря по ключу. Полезные функции и методы для работы со словарями.	2
<i>Практическое занятие</i> Работа с текстовыми файлами в Python. Добавление данных из файла в словарь. Функции и методы для работы со словарем. Решение прикладных задач.	2
<i>Практическое занятие</i> Решение прикладных задач по работе со словарем	2
<i>Практическое занятие</i> Изучение типа данных "множество". Отличия от других типов данных. Функции и методы для работы с множествами	2
<i>Практическое занятие</i> Работа с множествами. Решение прикладных задач	2

	<i>Практическое занятие</i> Json-формат. Десериализация данных	2
	<i>Практическое занятие</i> Json-формат. Десериализация данных	2
Промежуточная аттестация	Зачёт с оценкой Тестирование	1
Итого:		71

4.1.4. Материально-техническое обеспечение

Кабинет (лаборатория), мастерская	Оборудование и технические средства обучения
Мастерская за счет областного бюджета Программные решения для бизнеса	1 WiFi-точка доступа 2 аптечка 3 Интерфейсный кабель для подключения монитора 4 Кабель питания 5 Клавиатура 6 Компьютерная мышь 7 Компьютерный монитор 8 Компьютерный стул 9 Кондиционер 10 Маршрутизатор 11 Огнетушитель углекислотный ОУ-1 12 Офисный стол 13 Персональный компьютер в сборе 14 Проектор 15 сервер 16 Сетевой фильтр 17 Управляемый коммутатор

	18 Экран для проектора
	1 Веб-браузер Chrome
	2 Веб-браузер Opera
	3 ПО API Postman
	4 ПО IntelliJ IDEA
	5 ПО Microsoft Visual Studio Code
	6 ПО PyCharm
	7 ПО Python
	8 ПО для архивации WinRAR
	9 ПО для просмотра документов в формате PDF Adobe Reader DC
	10 ПО операционная система Microsoft Windows 10 Pro
	11 ПО офисный пакет Microsoft Office 2019
	12 ПО управления версиями Git
	13 Программное обеспечение 7-Zip

4.1.5. Кадровое обеспечение

Реализация образовательной программы обеспечивается педагогическими работниками колледжа, направление деятельности которых соответствует области профессиональной деятельности укрупненной группы специальностей 09.00.00 “Информатика и вычислительная техника” и имеющих стаж работы в данной профессиональной области не менее 3 лет. Педагогические работники, привлекаемые к реализации образовательной программы, получают дополнительное профессиональное образование по программам повышения квалификации, в том числе в форме стажировки в организациях, направление деятельности которых соответствует направлению обучения по специальностям укрупненной группы 09.00.00 “Информатика и вычислительная техника” не реже 1 раза в 3 года с учетом

расширения спектра профессиональных компетенций.

4.1.6. Организация образовательного процесса

Организация образовательного процесса регламентируется учебным планом и расписанием учебных занятий. Язык обучения (преподавания) - русский. Устанавливаются следующие формы учебных занятий: практическое занятие, самостоятельная работа. Для всех видов аудиторных занятий академический час устанавливается продолжительностью 45 минут. Организация образовательного процесса предусматривает применение активных и интерактивных форм проведения занятий в сочетании с внеаудиторной работой, а так же использование дистанционной формы обучения. Изучение программы модуля завершается зачетом с оценкой (фонд оценочных средств модуля представлен в приложении 1).

4.1.7. Информационное обеспечение обучения

Основная литература:

1. Гэддис Тони. Начинаем программировать на Python. 5-е изд., Пер. с англ.- СПб.: БХВ-Петербург, 2022.- 880с.:ил.

Дополнительная литература:

1. Д. Бейдер, Д. Эймос, Д. Яблонски, Ф. Хейслер Знакомство с Python .-СПб.: Питер, 2023.- 512с

Электронные и интернет-ресурсы:

1. PyCharm COMMUNITY EDITION. IDE для разработки на чистом Python [Электронный ресурс] URL: <https://www.jetbrains.com/pycharm/download/?section=windows> (дата обращения: 16.04.2025).

4.1.8. Контроль и оценка результатов освоения модуля

Результат освоения программы	Основные показатели оценки результата
ПК 1.1 Составление алгоритмов и запись их на языке программирования Python	Освоение основных алгоритмических конструкций языка Python, применение различных структур данных в решении задач
ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием	Экспертная оценка работоспособности программных модулей

Форма и вид аттестации по модулю:

Форма аттестации по модулю - Зачет. Задание для проведения зачета предполагает создание кода программ к базовым алгоритмам. Итоговое задание по модулю представлено в Приложении 1. По результатам выполнения зачетного задания выставляются оценки по четырехбалльной системе («отлично», «хорошо», «удовлетворительно», «неудовлетворительно»):

- «отлично» - разработан алгоритм
- «хорошо» - разработан алгоритм (присутствуют замечания по структуре программы, именованию переменных)
- «удовлетворительно» - необходимые умения работы с освоенным материалом в основном сформированы, программа имеет незначительные ошибки
- «не удовлетворительно» -необходимые умения не сформированы, программа не компилируется, присутствуют грубые ошибки .

5. Организационно-педагогические условия реализации программы

5.1. Материально-техническое обеспечение

Кабинет (лаборатория), мастерская	Оборудование и технические средства обучения
Мастерская за счет областного бюджета Программные решения для бизнеса	1 WiFi-точка доступа 2 аптечка 3 Интерфейсный кабель для подключения монитора 4 Кабель питания 5 Клавиатура 6 Компьютерная мышь 7 Компьютерный монитор 8 Компьютерный стул 9 Кондиционер 10 Маршрутизатор 11 Огнетушитель углекислотный ОУ-1 12 Офисный стол 13 Персональный компьютер в сборе 14 Проектор 15 сервер 16 Сетевой фильтр 17 Управляемый коммутатор 18 Экран для проектора 1 Веб-браузер Chrome 2 Веб-браузер Opera 3 ПО API Postman

	4 ПО IntelliJ IDEA
	5 ПО Microsoft Visual Studio Code
	6 ПО PyCharm
	7 ПО Python
	8 ПО для архивации WinRAR
	9 ПО для просмотра документов в формате PDF Adobe Reader DC
	10 ПО операционная система Microsoft Windows 10 Pro
	11 ПО офисный пакет Microsoft Office 2019
	12 ПО управления версиями Git
	13 Программное обеспечение 7-Zip

5.2. Кадровое обеспечение

Реализация образовательной программы обеспечивается дипломированными педагогическими кадрами, имеющими высшее профессиональное образование. Опыт деятельности в организациях соответствующей профессиональной сферы является обязательным для преподавателей, отвечающих за освоение обучающимся данной программы. Направление деятельности преподавателей должно соответствовать области профессиональной деятельности заявленной программы.

5.3. Организация образовательного процесса

Программа состоит из практико-ориентированных занятий, направленных на решение актуальных задач, характерных для профессии, которые позволяют организовать деятельностный подход и обеспечивают ознакомление учащихся с профессиональной компетенцией. Занятия проводятся очно.

5.4. Информационное обеспечение обучения

Основная литература:

1. Гэддис Тони. Начинаем программировать на Python. 5-е изд., Пер. с англ.- СПб.: БХВ-Петербург, 2022.- 880с.:ил.

Дополнительная литература:

1. Д. Бейдер, Д. Эймос, Д. Яблонски, Ф. Хейслер Знакомство с Python .-СПб.: Питер, 2023.- 512с

Электронные и интернет-ресурсы:

1. PyCharm COMMUNITY EDITION. IDE для разработки на чистом Python
[Электронный ресурс] URL:
<https://www.jetbrains.com/pycharm/download/?section=windows> (дата обращения:
16.04.2025).

6. Контроль и оценка результатов освоения программы

Результат освоения программы	Основные показатели оценки результата
ПК 1.1 Составление алгоритмов и запись их на языке программирования Python	Освоение основных алгоритмических конструкций языка Python, применение различных структур данных в решении задач
ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием	Экспертная оценка работоспособности программных модулей

Контроль и оценка результатов освоения программы:

Оценка качества освоения программы включает итоговую аттестацию слушателей. К итоговой аттестации допускаются слушатели, успешно прошедшие промежуточную аттестацию по модулям, предусмотренными учебным планом настоящей программы.

Форма итоговой аттестации по программе: Зачет с оценкой. Задание для проведения зачета представлено в Приложении 2.

Итоговая аттестация по программе: Зачет с оценкой, Зачет.

Фонд оценочных средств

Приложение №2

Фонд оценочных средств
для проведения промежуточной аттестации
по модулю: «Структуры данных Python»

г.Екатеринбург, 2025

Комплект оценочных средств

УСЛОВИЯ ВЫПОЛНЕНИЯ ЗАДАНИЯ

1. Место выполнения задания: мастерская «Программные решения для бизнеса»
2. Максимальное время выполнения задания: 1 академический час

ЗАДАНИЕ

Умение “видеть” алгоритмы и распознавать возможности их применения при решении задач. Поэтому вам необходимо реализовать алгоритмы, которые очень часто используются в практической деятельности

Список алгоритмов

1. Алгоритм перестановки двух чисел
2. Алгоритм нахождения минимального элемента в последовательности
3. Алгоритм нахождения максимального элемента в последовательности
4. Алгоритм нахождения суммы элементов последовательности
5. Алгоритм нахождения произведения элементов последовательности
6. Алгоритм вычисления факториала числа
7. Алгоритм нахождения всех цифр числа
8. Алгоритм определения простоты числа (простое число - то, которое делится только на себя и единицу)
9. Алгоритм замены одного символа в строке на другой
10. Алгоритм определения того, что введенный символ является числом
11. Алгоритм определения того, что введенный символ является буквой латинского алфавита
12. Алгоритм перебора элементов массива по индексам и изменения элемента по условию

Фонд оценочных средств

для проведения итоговой аттестации по программе
«Открой мир программирования с нуля на PYTHON»

Екатеринбург, 2025

Комплект оценочных средств

Комплект оценочных средств представляет собой тест, задания которого отражают следующие темы, изученные слушателями в модуле:

- Типы данных Python;
- Условия;
- Циклы while, for;
- Списки;
- Строки;
- Словари;
- Функции;
- Работа с текстовыми файлами

Условия выполнения задания:

1. Место выполнения задания: по выбору слушателя
2. Максимальное время выполнения задания: 40 минут.
3. Количество вопросов теста : 40
4. Максимальное количество баллов, которое может получить слушатель : 40

Список вопросов теста

1. С помощью какой функции в языке Python можно вывести информацию в терминал?
 - write()
 - print()
 - log()
 - out()

2. С помощью какой функции в языке Python можно ввести данные с клавиатуры в программу?

- `read()`
- `get()`
- `redline()`
- `input()`

3. Какие из способов объявления переменной являются корректными для языка Python?

- `var num = 2`
- `num = input()`
- `123_num = 456`
- `Very_big_num = 10**10`

4. Какого типа данных нет в языке программирования Python?

- Строка
- Целое число
- Число с плавающей точкой
- Есть все перечисленные

5. С помощью какой функции в языке Python можно преобразовать строку в целое число?

- `int()`
- `to_num()`
- `str()`

- list()

6. Что появится на экране в результате работы данной программы?

```
data=[1,2,3,'яблоко',[],0]
```

```
print(len(data))
```

В ответ укажите только соответствующее значение, без лишних пробельных символов.

7. С помощью каких функций или методов языка Python можно добавить элемент в список?

- add()
- insert()
- append()
- put()

8. Какие виды циклов существуют в языке Python?

- while
- do while
- repeat
- for

9. В какой структуре данных языка Python объекты представлены в виде пар «ключ-значение»?

- Словарь
- Список

- Строка
- Ничего из перечисленного

10. Сопоставьте функции, операторы и методы списков языка Python с их назначениями:

А. sort()	1. Возвращает сумму элементов списка
Б. del	2. Удаляет из списка элемент по указанному индексу
В. sum()	3. Добавляет элемент в конец списка
Г. append()	4. Сортирует элементы списка в порядке возрастания

11. С помощью какого метода можно обновить значение элемента словаря в языке Python?

- append()
- get()
- copy()
- update()

12. Что появится на экране в результате работы данной программы?

a=5

b=14

if a>0 and b<15:

 print(a+b)

elif a==b:

```
print(100)
else:
    print(0)
```

В ответ укажите только соответствующее значение, без лишних пробельных символов.

13.С помощью каких методов языка Python можно считывать информацию из текстового файла?

- read()
- write()
- readline()
- readlines()

14.С помощью какого ключевого слова в языке Python объявляются функции?

- func
- def
- if
- function

15.Какие из указанных конструкций позволяют в языке Python подключать библиотеки и их отдельные функции?

- import <название библиотеки>
- from <название библиотеки> import <название функции>
- import <название функции> from <название библиотеки>
- import <название функции>

16. Сопоставьте режим работы с файлом в языке Python с его назначением:

А) w	1) чтение
Б) w+	2) чтение и запись
В) a	3) запись
Г) r	4) добавление

17. С помощью какого метода в языке Python можно подсчитать количество вхождений указанного элемента в последовательность?

- sum()
- len()
- amount()
- count()

18. На вход программе подаются 10 натуральных чисел, каждое из них вводится через терминал с новой строки. Программа должна посчитать и вывести на экран произведение четных элементов среди введенных чисел. Если не было введено ни одного четного числа, то на экран выводится число -1.

Ниже приведен программный код с пронумерованными строчками для решения задачи, в котором содержатся **ошибки**. Найдите строки кода, в которых допущены ошибки и укажите в ответ их **номера** друг за другом, без пробелов, в порядке возрастания.

Например, если бы ошибки содержались в строчках № 1, 2 и 3, то в ответ нужно записать 123.

```

1 mult = 1
2 for i in range(20):
3     num = int(input())
4     if num % 2 == 0:
5         mult += num
6 if mult == 0:
7     print(-1)
8 else:
9     print(mult)

```

19. Найдите наименьшее значение переменной N, при котором представленная ниже программа выведет на экран число 24. В ответ укажите найденное число.

```

N = int(input())
summa = 0
for num in range(2, N, 2):
    if num % 4 == 0:
        summa += num
print(summa)

```

Перетяните в нужные ячейки подходящие по смыслу слова

20. «Почти готовая функция»

На вход функции `get_count_4` подается натуральное число n . Функция перебирает все целые числа в диапазоне от 1 до n включительно и возвращает количество кратных 4 чисел.

Ниже приведен код с частичной реализацией описанной функции. Допишите функцию, перетягивая в нужные ячейки подходящие команды.

```
def get_count_4(n) :
```

```
count=0

for I in range(1, n + 1):

    if i%4==0:

        count += 1

return count
```

21. Какая встроенная функция позволяет создать множество в языке программирования Python?

- list()
- int()
- set()
- dict()

22. Укажите верные утверждения относительно множеств в языке программирования Python:

- Множество является изменяемым типом данных
- Множество является неизменяемым типом данных
- В множестве могут храниться только изменяемые типы данных
- В множестве могут храниться только неизменяемые типы данных

23. Имеются 2 множества: `nums_1 = {1, 2, 9, 4, 8, 3}` и `nums_2 = {2, 5, 4, 3, 10, 9}`. Выберите верную конструкцию, чтобы в результате получилось новое множество с набором элементов 2 9 4 3.

- `nums_1.intersection(nums_2)`
- `nums_2.difference(nums_1)`
- `nums_1.union(nums_2)`
- `nums_1.append(nums_2)`

24. Какая встроенная функция позволяет создать кортеж в языке программирования Python?

- `set()`
- `tuple()`
- `dict()`
- `str()`

25. Укажите верные утверждения относительно кортежей в языке программирования Python:

- Кортеж является изменяемым типом данных
- К элементам кортежа можно обращаться по индексу
- При одинаковом количестве элементов кортежи занимают меньше места в памяти, чем списки
- Кортежи можно складывать с помощью операции

26. Выберите методы, которые можно применять к кортежам в языке программирования Python:

- `count()`

- `index()`
- `append()`
- `remove()`

27. В программе хранится двумерный список `nums`:

```
nums = [  
    [3, 1, 8],  
    [4, 2, 9],  
    [5, 7, 0]  
]
```

По каким **натуральным** индексам нужно обратиться к списку `nums`, чтобы получить доступ к числу 7?

В ответ укажите **сумму** индексов, без лишних пробельных символов.

28. Укажите неверные утверждения относительно вложенных структур данных в языке программирования Python:

- В словаре значением элемента может быть словарь
- Максимальный уровень вложенности любой структуры данных – 10
- С помощью циклов `for` нельзя перебрать элементы вложенных списков
- Вложенный словарь можно создать на основании другого словаря

29. С помощью какой конструкции можно создать список из натуральных чисел от 10 до 999 включительно в языке программирования Python?

- `[num for num in range(10, 999)]`
- `[num*2 for num in range(10, 1000)]`

- [for num in range(10, 999) append(num)]
- [num for num in range(10, 1000)]

30. Выберите верную общую форму генератора словарей в языке Python:

- {значение:ключ for переменная in последовательность if условие}
- {ключ:значение for переменная in последовательность if условие}
- {if условие for переменная in последовательность ключ:значение}
- {for переменная in последовательность if условие ключ:значение}

31. Какое условие нужно добавить в списочное выражение `nums = [num for num in range(10, 50)]`, чтобы в результате получился список `[14, 24, 34, 44]`?

- if i % 10 == 4
- if num % 4 == 0
- if i % 4 == 0
- if num % 10 == 4

32. Сопоставьте операторы и функции с их назначениями в языке Python:

Функция/оператор	Назначение
A. lambda	1. Оператор для создания анонимной функции
Б. map()	2. Сворачивает элементы последовательности в одно значение с использованием заданной функции
В. filter()	3. Позволяет убрать из указанной последовательности элементы, не подходящие под

	условие
Г. reduce()	4. Применяет указанную функцию ко всем объектам указанной последовательности

33. Укажите верные утверждения относительно лямбда-функций в языке программирования Python:

- У лямбда-функций есть конкретные имена
- У лямбда-функций нет конкретных имен
- Лямбда-функция должна возвращать значение
- Лямбда-функцию можно объявить с помощью оператора d

34. С помощью какой конструкции из строки `line = '12321412'` можно получить список `[1, 2, 3, 2, 1, 4, 1, 2]`, состоящих из целочисленных объектов?

- `list(map(int, line))`
- `list(filter(int, line))`
- `list(reduce(int, line))`
- `list(int, line)`

35. Сопоставьте операторы (в контексте обработки ошибок и исключений) и с их назначениями в языке Python:

Функция/оператор	Назначение
A. try	1. Блок кода, который вызывается при возникновении исключений

Б. except	2. Блок кода, который вызывается, если исключения не были обнаружены
В. else	3. Блок кода с некоторыми операциями, где могут возникнуть исключения
Г. finally	4. Блок кода, который выполняется в любом случае

36. Какие из указанных конфигураций операторов являются синтаксически верными в языке программирования Python:

- try-except
- try-finally
- try-except-else
- try-else

37. Какая из указанных ниже конструкций позволит обработать исключение, вызванное из-за некорректного ввода целочисленной переменной в коде `num = int(input())`?

- except ValueError:
- except SyntaxError:
- except ZeroDivisionError:
- except IndexError:

38. Выберите операторы, которые генерируют список аналогичный списку вида `a=[i for i in range(11)]`

- `a=list(range(11))`

- `a=tuple(range(11))`
- правильного ответа нет
- `a=[1 for i in range(11)]`
- `a=[0,1,2,3,4,5,6,7,8,9,10]`

39. Для вывода пар "ключ" - "значение" из словаря можно использовать следующие выражения:

- `for key,val in a.keys():print(key,val)`
- `for key,val in a.keys():print(key,a[key])`
- `for key,val in a.items():print(key,val)`
- `for key in a.keys():print(key,a[key])`
- `for key,val in a.items():print(key,a[key])`

40. Каким образом можно добавите в словарь `a={'a':5}` значение `{'b':6}`

- `a+={'b':6}`
- `a.update('b',6)`
- `a.append('b',6)`
- `a.update({'b':6})`
- `a.append({'b'},6)`