

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Дагестанский государственный университет»
Колледж

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП.04 Основы алгоритмизации и программирования

по программе подготовки специалистов среднего звена (ППССЗ) среднего
профессионального образования

<i>Специальность:</i>	09.02.07 Информационные системы и программирование
<i>Обучение:</i>	по программе базовой подготовке
<i>Уровень образования, на базе которого осваивается ППССЗ:</i>	основное общее образование
<i>Квалификация:</i>	программист
<i>Форма обучения:</i>	очная

Рабочая программа дисциплины «Основы алгоритмизации и программирования» разработана на основе требований Федерального государственного образовательного стандарта (далее ФГОС) для среднего профессионального образования (СПО) по специальности 09.02.07 «Информационные системы и программирование» для реализации основной профессиональной образовательной программы СПО на базе основного общего образования с получением среднего общего образования.

Организация-разработчик: Колледж федерального государственного бюджетного образовательного учреждения высшего образования «Дагестанский государственный университет».

Разработчики:

Магомедова П. Р.- к.ю.н., доцент, зав кафедрой общепрофессиональных дисциплин Колледжа ДГУ.

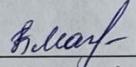
Хангишиева А.Х. – преподаватель кафедры общепрофессиональных дисциплин Колледжа ДГУ.

Рецензент:

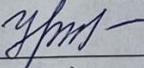
Исмиханов З.Н.- к.э.н., доцент, зав. кафедрой информационных систем и технологий программирования факультета ИИТ ДГУ.

Рабочая программа дисциплины рассмотрена и рекомендована к утверждению на заседании кафедры общепрофессиональных дисциплин Колледжа ДГУ.

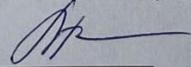
Протокол № 7 от « 31 » сентября 2022 г.

Зав. кафедры  /Магомедова П.Р.
подпись Фамилия И.О.

Утверждена на заседании учебно-методического совета колледжа ДГУ

Ст. методист  /Шамсутдинова У.А.
подпись Фамилия И.О.

Рабочая программа дисциплины согласована с учебно-методическим управлением

« 31 » 03 2022 г. 
подпись

СОДЕРЖАНИЕ

1. Паспорт программы учебной дисциплины «Основы алгоритмизации и программирования»
2. Структура и содержание дисциплины
3. Условия реализации дисциплины
4. Контроль и оценка результатов освоения дисциплины

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Основы алгоритмизации и программирования

1.1. Область применения программы

Рабочая программа дисциплины является частью основной профессиональной образовательной программы в соответствии с ФГОС по специальности СПО 10.02.05. Обеспечение информационной безопасности автоматизированных систем для очного обучения студентов, имеющих основное общее образование, по программе базовой подготовки.

Рабочие программы дисциплин, адаптированные для обучения лиц с ограниченными возможностями здоровья, разрабатываются с учетом конкретных ограничений здоровья лиц, зачисленных в колледж, и утверждаются в установленном порядке.

1.2. Место дисциплины в структуре основной профессиональной образовательной программы

Учебная дисциплина «Основы алгоритмизации и программирования» относится к общепрофессиональным дисциплинам профессионального цикла ППСЗ

1.3. Цели и задачи дисциплины – требования к результатам освоения дисциплины

Содержание программы учебной дисциплины «Основы алгоритмизации и программирования» направлено на достижение следующих целей:

- изучение и применение на практике основных методов и средств обработки, хранения, передачи и поиска информации с использованием различных программных средств,

Задачи учебной дисциплины «Основы алгоритмизации и программирования» в профессиональной деятельности:

1 Понимать сущность и социальную значимость своей будущей профессии, обладать высокой мотивацией к выполнению профессиональной деятельности в области обеспечения информационной безопасности

2 Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество

3 Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития

4 Использовать информационно-коммуникационные технологии в профессиональной деятельности

5 Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации

6 Ориентироваться в условиях частой смены технологий в профессиональной деятельности

7 Владеть основными методами и средствами разработки программного обеспечения

8 Решать частные технические задачи, возникающие при проведении всех видов плановых и внеплановых контрольных проверок, при аттестации объектов, помещений, программ, алгоритмов.

Освоение содержания учебной дисциплины «Основы алгоритмизации и программирования» обеспечивает достижение студентами следующих результатов:

Общие компетенции:

ОК-01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК-02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК-03. Планировать и реализовывать собственное профессиональное и личностное развитие

Профессиональные компетенции:

ПК-2.1. Осуществлять установку и настройку отдельных программных, программно-аппаратных средств защиты информации.

ПК-2.2. Обеспечивать защиту информации в автоматизированных системах отдельными программными, программно-аппаратными средствами.

ПК-2.3.Осуществлять тестирование функций отдельных программных и программно-аппаратных средств защиты информации.

ПК-2.4.Осуществлять обработку, хранение и передачу информации ограниченного доступа.

ПК-2.6.Осуществлять регистрацию основных событий в автоматизированных (информационных) системах, в том числе с использованием программных и программно-аппаратных средств обнаружения, предупреждения и ликвидации последствий компьютерных атак.

В результате освоения дисциплины обучающийся должен знать:

- типы данных, базовые конструкции изучаемых языков программирования;
- интегрированные среды изучаемых языков программирования.

В результате освоения дисциплины обучающийся должен уметь:

- использовать языки программирования высокого уровня;
- работать в интегрированной среде программирования.

2. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов
Максимальная учебная нагрузка (всего)	196
Обязательная аудиторная учебная нагрузка (всего)	154
в том числе:	
теоретическое обучение	78
лабораторные работы	-
практические занятия	76
контрольные работы	4
курсовой проект	-
консультация	1
экзамен	4
дифференцированный зачет	3
Самостоятельная работа обучающегося (всего)	41
в том числе:	
самостоятельная работа над курсовой работой	
внеаудиторная самостоятельная работа	
Промежуточная аттестация в форме – экзамен, дифференцированный зачет	

2.2. Тематический план и содержание дисциплины Основы алгоритмизации и программирования

Наименование разделов и тем	Содержание учебного материала, лекций, лабораторные и практические занятия, самостоятельная работа обучающихся, курсовая работа (проект)	Объем часов	Формы и методы контроля и оценки результата в обучения	
1	2	3	4	
Раздел 1. Введение в программирование				
Тема 1.1. Языки программирования.	Лекции	8		
	1	Развитие языков программирования	2	
	2	Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы.	2	
	3	Жизненный цикл программы. Программа. Программный продукт и его характеристики.	2	
	4	Основные этапы решения задач на компьютере.	2	
	Практические занятия:		2	
	Создание приложения с выводом текста на консоль.			Устный опрос, Оценка фактического материала
	Самостоятельная работа обучающихся: * – подготовка презентаций, рефератов по теме занятия. Примерная тематика: – эволюция языков программирования; – машинно-ориентированные языки; – процедурные языки программирования; – логические языки программирования; – интерпретаторы и компиляторы; – интегрированные среды; – визуальные среды программирования; – встроенные языки программирования; – технология RAD; – CASE-технологии		2	Оценка умения, анализ решения задач, тестирование
Тема 1.2. Типы данных	Лекции	2		
	1	Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных.	2	
	Практические занятия 2			
	Создание консольного приложения с установкой типа данных и вывода результата		2	Устный опрос, Оценка фактического материала

	Самостоятельная работа обучающихся:* - домашние практические работы на построение алгоритмов решения простейших задач с использованием данных различных типов		3	Оценка умения, анализ решения задач
Раздел 2. Основные конструкции языков программирования				
Тема 2.1. Операторы языка программирования	Лекции		30	
	1	Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор.	4	
	2	Условный оператор. Оператор выбора.	6	
	3	Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы.	6	
	4	Массивы. Двумерные массивы. Строки. Стандартные процедуры и функции для работы со строками.	6	
	5	Структурированный тип данных – множество. Операции над множествами.	4	
	6	Комбинированный тип данных – запись. Файлы последовательного доступа. Файлы прямого доступа	4	
	Практические занятия			
	Создание программ линейной структуры.		4	Устный опрос, Оценка фактического материала
	Создание программ разветвляющейся структуры.		4	Устный опрос, Оценка фактического материала
	Создание программ циклической структуры.		4	Устный опрос, Оценка фактического материала
	Создание программ на преобразование элементов одномерного массива		4	Устный опрос, Оценка фактического материала
	Создание программ на преобразование элементов двумерного массива		4	Устный опрос, Оценка фактического материала
	Составления программ с применением множеств для решения задач.		4	Устный опрос, Оценка фактического материала
Составление программ на создание текстового файла для решения задач		4	Устный опрос,	

			Оценка фактического материала	
	Создание программ для работы с каталогами и файлами	4	Устный опрос, Оценка фактического материала	
	Самостоятельная работа обучающихся: * -домашние практические работы на программирование алгоритмов для консольного приложения.	2	Оценка умения, анализ решения задач	
Раздел 3. Структурное и модульное программирование				
Тема 3.1. Процедуры и функции	Лекции		4	
	1	Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций.	2	
	2	Рекурсия. Программирование рекурсивных алгоритмов.	2	
	Практические занятия:		12	
	Программирование алгоритмов с использованием процедур и функций.		4	Устный опрос, Оценка фактического материала
	Организация функций.		4	Устный опрос, Оценка фактического материала
	Применение рекурсивных функций.		4	Устный опрос, Оценка фактического материала
	Самостоятельная работа обучающихся:* -домашние практические работы на создание приложения для отображения случайных чисел		2	Оценка умения, анализ решения задач
Тема 3.2. Структуризация в программировании	Лекции		4	
	1	Основы структурного программирования.	2	
	2	Методы структурного программирования.	2	
	Самостоятельная работа обучающихся:* - подготовка сообщений, рефератов по теме занятия. Примерная тематика: – модульный принцип программирования. – подпрограммы. – принципы проектирования программ "сверху-вниз" и "снизу-вверх"			Оценка умения, анализ решения задач, тестирование
Тема 3.3. Модульное программирование	Лекции		2	
	1	Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы. Стандартные модули	2	
	Практические занятия:		8	

	Подключение модулей. Создание собственного модуля.	4	Устный опрос, Оценка фактического материала	
	Использование модулей для консольного приложения	4	Устный опрос, Оценка фактического материала	
	Самостоятельная работа обучающихся:* -домашние практические работы на создание приложения с применением стандартных модулей	2	колоквиум	
Раздел 4. Указатели и динамические структуры				
Тема 4.1. Указатели.	Лекции		6	
	1	Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных.	2	
	2	Структуры данных на основе указателей.	2	
	3	Задача о стеке	2	
	Самостоятельная работа обучающихся:* подготовка презентации, реферата по теме: переменные ссылки		2	Оценка умения, анализ решения задач
Раздел 5. Объектно-ориентированное программирование				
Тема 5.1. Основные принципы объектно-ориентированного программирования (ООП)	Лекции		8	
	1	История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.	2	
	2	Основные принципы ООП: инкапсуляция, наследование, полиморфизм.	2	
	3	Классы объектов. Компоненты и их свойства.	2	
	4	Событийно-управляемая модель программирования. Компонентно-ориентированный подход.	2	
	Практические занятия:		4	
	Программирование простейших классов.		2	Устный опрос, Оценка фактического материала
	Классы ООП: виды, назначение, свойства, методы, события.		2	Устный опрос, Оценка фактического материала
Самостоятельная работа обучающихся:* - домашние практические работы на создание проекта с использованием компонентов для работы с текстом		2	Оценка умения, анализ решения задач	
Тема 5.2. Интегрированная среда	Лекции		2	
	1	Требования к аппаратным и программным средствам интегрированной среды	2	

разработчика.		разработчика. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.		
	Самостоятельная работа обучающихся:* -домашние практические работы на изучение интегрированной среды разработчика		2	Оценка умения, анализ решения задач
Тема 5.3. Визуальное событийно-управляемое программирование	Лекции		4	
	1	Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение. Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектом через свойства.	2	
	2	События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий.	2	
	Практические занятия:		4	
	Введение в разработку графических пользовательских интерфейсов		4	Устный опрос, оценка фактического материала
	Самостоятельная работа обучающихся: * -домашние практические работы с использованием компонентов ввода и отображения чисел, дат и времени.		2	Тестирование, оценка умения, анализ решения задач,
Тема 5.4. Разработка оконного приложения	Лекции		4	
	1	Разработка функционального интерфейса приложения. Создание интерфейса приложения.	2	
	2	Разработка функциональной схемы работы приложения. Разработка приложения.	2	
	Практические занятия:			
	Разработка интерфейса приложения.		4	Устный опрос, Оценка фактического материала
Самостоятельная работа обучающихся:* -домашние практические работы с использованием кнопочных компонентов.		2	Оценка умения, анализ решения задач	
Тема 5.5. Этапы разработки	Лекции		2	
	1	Разработка приложения. Проектирование объектно-	2	

приложений		ориентированного приложения Создание интерфейса пользователя. Тестирование, отладка приложения.		
	Практические занятия:		6	
		Разработка подсистемы регистрации и авторизации для автоматизированной информационной системы.	4	Устный опрос, оценка фактического материала
		Разработка минимального функционала автоматизированной информационной системы.	2	Устный опрос, оценка фактического материала
		Самостоятельная работа обучающихся:* -домашние практические работы с использованием компонентов стандартных диалогов	2	тестирование
Тема 5.6. Иерархия классов.	Лекции		2	
	1	Классы ООП: виды, назначение, свойства, методы, события. Перегрузка методов. Тестирование и отладка приложения. Решение задач	2	
	Практические занятия:		2	
		Реализация иерархии наследования классов.	2	Устный опрос, оценка фактического материала
		Самостоятельная работа обучающихся:* -домашние практические работы с использованием перегрузки методов.	2	Оценка умения, анализ решения задач
Примерная тематика курсовой работы(проекта) <i>если предусмотрены</i>				
<i>Лекционные занятия</i>			78	
<i>Практические занятия</i>			76	
<i>Самостоятельная работа</i>			41	
Итого			196	
Форма контроля			Экзамен Дифференцированный зачет	

**в перечень вопросов для самостоятельного изучения включаются вопросы, не вошедшие в раздел семинарского и практического занятий.*

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ

3.1. Требования к минимальному материально-техническому обеспечению

Реализация программы дисциплины требует наличия учебного кабинета.

Оборудование учебного кабинета:

- посадочные места по количеству обучающихся;
- рабочее место преподавателя; -
- комплект учебно-наглядных пособий;

Технические средства обучения:

- компьютер по количеству обучающихся с лицензионным программным обеспечением и с доступом к сети Интернет.
- проектор;
- интерактивная доска;

Программное обеспечение:

- Интегрированная среда программирования (Python, Microsoft Visual Studio, и т.п.)
- Операционные системы Windows 7, SuseLinux 10.
- Программы настройки и оптимизации операционной системы: NortonUtilities, SystemUtilities, Sandra, SystemMechanic.
- Архиваторы: WinRar, WinZip, ZipMagic, WinAce.
- Антивирусные средства: Drweb, Avp.
- Программы для работы с изображением: ACDSee, AcrobatReader.
- Программы для работы с Internet и электронной почтой: EtypeDialer, GetRight, TheBat!, Ace FTP, Opera, ICQ. 10
- Программы-оболочки: FAR manager 2.0, Volkov Commander.
- MS Office 2007/2010.
- СУБД ACCESS.
- СерверБД MS SQL 2005.
- Программы для тестирования аппаратных устройств ПЭВМ.
- Текстовые процессоры / редакторы: Word, LaTeX.
- Программы для создания компьютерных презентаций MS PowerPoint

3.2. Информационное обеспечение обучения

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы.

Основная литература:

1. *Кудрина, Е. В.* Основы алгоритмизации и программирования на языке C# : учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2022. — 322 с. — (Профессиональное образование). — ISBN 978-5-534-10772-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/494914>
2. *Трофимов, В. В.* Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2022. — 137 с. — (Профессиональное образование). — ISBN 978-5-534-07321-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/493261>

Дополнительная литература:

1. Гниденко, И. Г. Технология разработки программного обеспечения : учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2022. — 235 с.
2. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C# : учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2022. — 322 с.
3. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное

- пособие для среднего профессионального образования / Д. Ю. Федоров. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2022. — 210 с.
4. Чернышев, С. А. Основы программирования на Python : учебное пособие для среднего профессионального образования / С. А. Чернышев. — Москва : Издательство Юрайт, 2022. — 286 с

Электронные ресурсы:

1. Официальный сайт Python [Электронный ресурс]:- Режим доступа: <https://www.python.org>
2. Руководство по языку программирования Python – Metanit [Электронный ресурс]:- Режим доступа: <https://metanit.com> > python > tutorial
3. Самоучитель Python | Python 3 для начинающих и чайников [Электронный ресурс]:- Режим доступа: <https://pythonworld.ru> > samouchitel-python
4. Учебник Python 3.1 [Электронный ресурс]:- Режим доступа: <https://ru.wikibooks.org>

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Контроль и оценка результатов освоения дисциплины осуществляется преподавателем в процессе проведения практических занятий и лабораторных работ, тестирования, а также выполнения обучающимися индивидуальных заданий, проектов, исследований.

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля результатов обучения
<p>В результате освоения дисциплины обучающийся должен знать:</p> <ul style="list-style-type: none"> - интегрированные среды изучаемых языков программирования - обнаруживать и устранять неисправности системы защиты информации автоматизированной системы согласно документации (06.033 А/01.5) - типы данных, базовые конструкции изучаемых языков программирования; 	<ul style="list-style-type: none"> - Оценка в рамках текущего контроля результатов деятельности обучающихся при выполнении: <ul style="list-style-type: none"> - проверка конспектов лекций; - тестирование; - самостоятельная работа; - выполнение индивидуальных практических заданий; - индивидуальный и фронтальный опрос в ходе аудиторных занятий; - защита практической работы
<p>В результате освоения дисциплины обучающийся должен уметь:</p>	
<ul style="list-style-type: none"> - использовать языки программирования высокого 	<ul style="list-style-type: none"> - Оценка в рамках текущего

уровня; - работать в интегрированной системе программирования;	контроля результатов деятельности обучающихся при выполнении: – проверка конспектов лекций; – тестирование; – самостоятельная работа; – выполнение индивидуальных Практических/лабораторных заданий; – индивидуальный и фронтальный опрос в ходе аудиторных занятий; – защита практической работы
--	---

Перечень зачетно-экзаменационных вопросов

1. Основные понятия алгоритмизации.
2. Способы описания алгоритмов.
3. Способ описания алгоритмов записи на естественном языке?
4. Свойства алгоритмов.
5. Линейные алгоритмы.
6. Алгоритмы с ветвлениями.
7. Циклические алгоритмы.
8. Базовые конструкции структурного программирования.
9. Цикл.
10. Цикл с предусловием.
11. Цикл с постусловием
12. Компилятор
13. Интерпретатор
14. Байт-код.
15. Программа, которые преобразуют исходные тексты программ, написанные на языке программирования высокого уровня, в программу на машинном языке, «понятную» компьютеру?
16. Языки программирования, ориентированные на команды процессора и учитывающие его особенности?
17. Какие языки не относятся к языкам программирования:
18. Переменные.
19. Операция присваивания.
20. Типы данных.
21. Арифметические операции.
22. В программе связь между данными и переменными устанавливается с помощью знака?
23. Булевы типы.
24. В каких условиях составное высказывание А И В может быть истинным?
25. Объединение двух высказываний с помощью оборота «если..., то...» называется:
26. Таблица, содержащая все возможные значения логического выражения, называется:
27. Как кодируется логическая переменная, принимающая значение «ЛОЖЬ»?
28. Символьный тип данных.
29. Оператор выбора.
30. Конструкция if...else. Синтаксис конструкции.
31. Цикл. For. Синтаксис конструкции.
32. Цикл с предусловием (цикл пока)?
33. Цикл с пост условием(цикл до)?

34. Оператор?
35. Выражение?
36. Цикл с параметром?
37. Подпрограмма (функция)?
38. Объективно-ориентированное программирование?
39. Программа-компилятор выполняет:
40. Отличительными чертами языков программирования высокого уровня являются:

41. Языки высокого уровня :
42. Среда разработки программного обеспечения – это?
43. Языками программирования являются:
44. Список (list).
45. Создание списка.
46. Добавление элемента в список
47. Удаление элемента из списка.
48. Массив. Задание значений элементов массива.
49. Массив. Удаление элемента.
50. Структурированный тип данных: строки.
51. Текстовые файлы.
52. Операции с файлом.
53. Принципы структурного программирования.
54. Инструкция для ввода значений переменных с клавиатуры:
55. Логические значения представляют одно из двух значений:
56. Это простейший вариант цикла. Он выполняет некоторые действия, после чего определяет, не надо ли их выполнить еще раз. «Определяет» путем проверки указанного программистом условия.
57. Оператор break
58. Оператор continue
59. Тело цикла — это собственно те команды, которые находятся внутри цикла.
60. Строки — это последовательности символов.
61. Строковый тип данных называется ?
62. Чтобы получить длину строки, используйте функцию?
63. Процедура .
64. Глобальная переменная
65. Локальная переменная.
66. Упорядоченная и изменяемая коллекция?
67. Упорядоченная и неизменяемая коллекция?
68. Неупорядоченная, неизменяемая и неиндексированная коллекция?
69. Сборник, упорядоченный и изменяемый?
70. Набор объектов одного типа под общим именем?
71. Каждый объект имеет свой номер, с помощью которого мы обращаемся к этому объекту?
72. Объявление массива в Python известного размера.
73. Создать множество.
74. Добавление одного элемента в множество
75. Добавление нескольких элементов в множество
76. Удаление элементов из множеств
77. Метод, который применяют в тех случаях, когда нужно удалить из множества конкретный элемент?
78. Комбинированный тип данных
79. Набор данных, сохраненный в виде последовательности битов на компьютере?
80. В Python существует два типа файлов:
81. Любую операцию с файлом можно разбить на три крупных этапа:
82. В Python есть встроенная функция _____ с ее помощью можно открыть любой файл на компьютере
83. Закрытие файла.
84. Функция _____ используется для чтения содержимого файла после открытия его в режиме

чтения (r).

85. Функция_____используется для записи в файлы Python, открытые в режиме записи.
86. Используем turtle для написания текста.
87. turtle .Установить цвет , чтобы создать цветной текст:
88. Команды для рисования turtle .
89. Фоновый цвет кнопки
90. Устанавливает размер окна.
91. Устанавливает текст окна.
92. Сделает элемент видимым.
93. Использовать функцию_____, чтобы остановить выполнение программы на заданное время в секундах.
94. Графическая библиотека Python, которая предназначена для создания программ с оконным интерфейсом?
95. Для работы с наборами данных Python предоставляет такие встроенные типы как?
96. Инструкция для ввода значений переменных с клавиатуры:
97. Логические значения представляют одно из двух значений:
98. Это простейший вариант цикла. Он выполняет некоторые действия, после чего определяет, не надо ли их выполнить еще раз. «Определяет» путем проверки указанного программистом условия.
99. Тело цикла
- 100.Чувствителен ли Python к регистру ?