

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г. ШУХОВА**
ФИЛИАЛ ФГБОУ ВО «БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г. ШУХОВА»
в г. НОВОРОССИЙСКЕ
(**НФ БГТУ им. В.Г. Шухова**)

УТВЕРЖДАЮ
Директор филиала
И.В. Чистяков
« 20 » августа 2021 г.



РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ОПЦ.04 Основы алгоритмизации и программирования

наименование дисциплины

Специальность: *09.02.07. Информационные системы и программирование*

Квалификация: *специалист по информационным системам*

Форма обучения: *очная*

Срок обучения: *3 года 10 месяцев*

Новороссийск– 2021

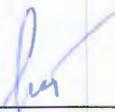
Рабочая программа разработана на основе:

- требований Федерального государственного образовательного стандарта среднего общего образования, приказ Министерства образования и науки от 9 декабря 2016 г. № 1547 с изменениями и дополнениями (зарегистрировано в Минюсте РФ 26 декабря 2016 г., N44936)

- учебного плана программы подготовки специалистов среднего звена 09.02.07 Информационные системы и программирование, входящей в укрупненную группу специальностей 09.00.00 Информатика и вычислительная техника.

Составитель: к.физ.мат.н., доц

ученая степень и звание



подпись

О.В.Мкртычев

инициалы, фамилия

Рабочая программа обсуждена на заседании кафедры

Технических дисциплин

название кафедры

«17» августа 2021 г., протокол № 6

Заведующий кафедрой: д.т.н., проф.

ученая степень и звание



подпись

Г.Ю.Ермоленко

инициалы, фамилия

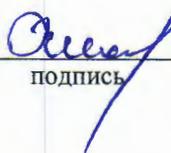
Программа одобрена научно-методическим советом филиала

« 19 » августа 2021 г., протокол № 3

Председатель:

к.ф.н., доц.

ученая степень и звание



подпись

И.В.Чистяков

инициалы, фамилия

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОПЦ. 04 Основы алгоритмизации и программирования

1.1. Область применения программы

Рабочая программа учебной дисциплины является частью основной профессиональной образовательной программы в соответствии с ФГОС СПО по специальности 09.02.07 Информационные системы и программирование, входящей в состав укрупненной группы специальностей 09.00.00 Информатика и вычислительная техника. Рабочая программа учебной дисциплины может быть использована в дополнительном профессиональном образовании (в программах повышения квалификации и переподготовки) и профессиональной подготовке работников в области Информационных систем и программирования

1.2. Место дисциплины в структуре основной профессиональной образовательной программы

ОПЦ –общепрофессиональный цикл

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

В результате освоения дисциплины обучающийся должен уметь:

Разрабатывать алгоритмы для конкретных задач.

Использовать программы для графического отображения алгоритмов.

Определять сложность работы алгоритмов.

Работать в среде программирования.

Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Оформлять код программы в соответствии со стандартом кодирования.

Выполнять проверку, отладку кода программы. .

В результате освоения дисциплины обучающийся должен знать:

Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.

Эволюцию языков программирования, их классификацию, понятие системы программирования.

Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.

Подпрограммы, составление библиотек подпрограмм

Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения

1.4. Общие и профессиональные компетенции, формируемые в ходе освоения учебной дисциплины

Программа учебной дисциплины в соответствии с ФГОС способствует формированию следующих общих и профессиональных компетенций на основе применения активных методов обучения:

Код ОК	Наименование компетенции	Методы обучения
ОК 1.	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	Задания проблемного характера;
ОК 2.	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности	Задания проблемного характера;
ОК 9.	Использовать информационные технологии в профессиональной деятельности	Задания проблемного характера;
ПК 2.5	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования	Задания проблемного характера;

1.5. Количество часов на освоение рабочей программы учебной дисциплины максимальной учебной нагрузки обучающегося 168 часов, в том числе: обязательной аудиторной учебной нагрузки обучающегося 140 часов; самостоятельной работы обучающегося 18 часов

1.6. Использование в рабочей программе часов вариативной части

Учебным планом не предусмотрено

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Всего часов	Семестр № 3	Семестр № 4
Максимальная учебная нагрузка (всего)	168	88	80
Обязательная аудиторная учебная нагрузка (всего)	140	68	72
в том числе:			
лекции, уроки	70	34	36
практические занятия	70	34	36
лабораторные занятия			
семинарские занятия			
контрольные работы			
курсовая работа (проект)			
Внеаудиторная (самостоятельная) учебная работа обучающегося	18	12	6
Консультации	2		2
Промежуточная аттестация в форме <i>3 семестр – экзамен</i> <i>4 семестр – дифференцированный зачет</i>	8	8	

2.2. Тематический план и содержание учебной дисциплины

Наименование разделов и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся, курсовая работа (проект) (если предусмотрены)	Объем часов	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
Раздел 1.	<i>Введение в программирование</i>	18	ОК 1 ОК 2 ОК 9
Тема 1.1. Языки программирования	1. Развитие языков программирования. 2. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы. 3. Жизненный цикл программы. Программа. Программный продукт и его характеристики. 4. Основные этапы решения задач на компьютере.	6	
	В том числе практических занятий и лабораторных работ Знакомство со средой программирования.	4	
Тема 1.2. Типы данных	Содержание учебного материала	4	
	1. Типы данных. Простые типы данных. Коллекции. Числа. Строки. Списки. Кортежи. Словари		
	В том числе практических занятий и лабораторных работ Составление программ линейной структуры. Составление программ разветвляющейся структуры. Составление программ циклической структуры	4	
Раздел 2.	Содержание учебного материала	20	ОК 1 ОК 2 ОК 9
Тема 2.1. Операторы языка программирования	1. Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор.	6	

	Промежуточная аттестация		
	2. Условный оператор. Оператор выбора.		ОК 1 ОК 2 ОК 9
	3. Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы.		
	4. Массивы. Двумерные массивы. Строки. Стандартные процедуры и функции для работы со строками.		
	5. Структурированный тип данных - множество. Операции над множествами.		
	6. Комбинированный тип данных - запись. Файлы последовательного доступа. Файлы прямого доступа		
	В том числе практических занятий и лабораторных работ Обработка одномерных массивов. Обработка двумерных массивов. Работа со строками. Работа с данными типа множество. Файлы последовательного доступа. Типизированные файлы. Нетипизированные файлы.	14	
Раздел 3.	Содержание учебного материала	36	ОК 1 ОК 2 ОК 9 ПК 2.5
Тема 3.1. Процедуры и функции	1. Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций.	4	
	2. Рекурсия. Программирование рекурсивных алгоритмов. В том числе практических занятий и лабораторных работ Организация процедур. Организация функций. Применение рекурсивных функций.	10	
Тема 3.2. Структуризация в программировании	Содержание учебного материала	6	
	1. Основы структурного программирования. Методы структурного программирования. В том числе практических занятий и лабораторных работ		

Тема 3.3. Модульное программирование	Содержание учебного материала		
	1. Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы.	4	ОК 1 ОК 2 ОК 9 ПК 2.5
	2. Стандартные модули. В том числе практических занятий и лабораторных работ	6	
	Программирование модуля. Создание библиотеки подпрограмм.		
Раздел 4	<i>Основные конструкции языков программирования</i>	12	
Тема 4.1 Указатели.	Содержание учебного материала	6	ОК 1 ОК 2 ОК 9 ПК 2.5
	1. Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных.		
	2. Структуры данных на основе указателей. 3. Задача о стеке. В том числе практических занятий и лабораторных работ		
	Использование указателей для организации связанных списков.	6	
Раздел 5	Содержание учебного материала	74	
Тема 5.1 Основные принципы объектно-ориентированного программирования (ООП)	1. История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.	10	ОК 1 ОК 2 ОК 9 ПК 2.5
	2. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.		
	3. Классы объектов. Компоненты и их свойства.		
	4. Событийно-управляемая модель программирования. Компонентноориентированный подход. В том числе практических занятий и лабораторных работ	10	
	Классы ООП: виды, назначение, свойства, методы, события. Объявления класса. Создание наследованного класса. Программирование приложений.		

Тема 5.2 Интегрированная среда разработчика.	Содержание учебного материала	6	
	1. Требования к аппаратным и программным средствам интегрированной среды разработчика. 2. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов.		
	3. Панель компонентов и их свойства. Окно кода проекта.	2	
	4. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.		
	5. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.		
	6. Настройка среды и параметров проекта.		
	В том числе практических занятий и лабораторных работ Изучение интегрированной среды разработчика		
Тема 5.3. Визуальное событийно-управляемое программирование	Содержание учебного материала	6	ОК 1 ОК 2 ОК 9 ПК 2.5
	1. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.		
	2. Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектом через свойства.		
	3. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий.		
	В том числе практических занятий и лабораторных работ Создание проекта с использованием компонентов для работы с текстом. Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий. Создание проекта с использованием кнопочных компонентов. Создание проекта с использованием компонентов стандартных диалогов и системы меню. Разработка функциональной схемы работы приложения. Разработка оконного приложения с несколькими формами. Разработка игрового приложения.	10	

Тема 5.4 Разработка оконного приложения	Содержание учебного материала	8	
	1. Разработка функционального интерфейса приложения. Создание интерфейса приложения.		ОК 1 ОК 2 ОК 9 ПК 2.5
	2. Разработка функциональной схемы работы приложения.		
	3. Разработка игрового приложения.		
	В том числе практических занятий и лабораторных работ Создание процедур обработки событий. Компиляция и запуск приложения. Разработка интерфейса приложения. Тестирование, отладка приложения.	2	
Тема 5.5 Этапы разработки приложений	Содержание учебного материала		
	1. Разработка приложения.		ОК 1 ОК 2 ОК 9 ПК 2.5
	2. Проектирование объектно-ориентированного приложения.	8	
	3. Создание интерфейса пользователя.		
	4. Тестирование, отладка приложения.		
	В том числе практических занятий и лабораторных работ Создание процедур обработки событий. Компиляция и запуск приложения. Разработка интерфейса приложения. Тестирование, отладка приложения.	4	
Тема 5.6 Иерархия классов.	Содержание учебного материала		
	1. Классы ООП: виды, назначение, свойства, методы, события.		ОК 1 ОК 2 ОК 9 ПК 2.5
	2. Перегрузка методов.	6	
	3. Тестирование и отладка приложения.		
	4. Решение задач		
	В том числе практических занятий и лабораторных работ Перегрузка методов.	2	
Промежуточная аттестация - экзамен		8	
ВСЕГО		168	

Для характеристики уровня освоения учебного материала используются следующие обозначения:

1. - ознакомительный (узнавание ранее изученных объектов, свойств);
2. - репродуктивный (выполнение деятельности по образцу, инструкции или под руководством)
3. - продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач)

3. УСЛОВИЯ РЕАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1. Требования к материально-техническому обеспечению

Наименование оборудованных учебных кабинетов, объектов для проведения практических занятий, объектов физической культуры и спорта с перечнем основного оборудования	Адрес (местоположение) учебных кабинетов, объектов для проведения практических занятий, объектов физической культуры и спорта (с указанием площади и номера помещения в соответствии с документами бюро технической инвентаризации)
<p>Лаборатория разработки программного обеспечения № 413 для проведения учебных занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Оснащен специализированной мебелью, кондиционером, персональными компьютерами с выходом в Интернет и обеспечением доступа в электронную информационно-образовательную среду филиала, телевизором, веб-камерой, графическим планшетом, программным пакетом Microsoft Windows 10 (ОЕМ лицензия), Microsoft Office Стандартный 2007 (академическая лицензия № 49190957 от 20.10.2011); Dr. Web Security Space 12 - сублицензионный договор 490 от 10.08.2021; браузеры Google Chrome, Internet Explorer, Zoom, Sumatra PDF, 7Zip, Eclipse IDE for JAVA EED Developers, .NetFrameworkJDK8, Microsoft SQL Server Express Edition, Microsoft Visual Studio, My SQL Installer, NetBeans, SQLServer Management Studio, Android Studio, IntelliJDEA – свободно распространяемое ПО согласно условиям лицензионного соглашения</p>	<p>353919, Краснодарский край, г. Новороссийск, ул. Мысхакское шоссе, дом № 75, аудитория № 413 35,8 кв.м., этаж 4, помещение 413</p>
<p>Учебное помещение 407 для проведения групповых и индивидуальных консультаций, текущего контроля, самостоятельной работы. Оснащен специализированной мебелью, персональным компьютером с выходом в Интернет и обеспечением доступа в электронную информационно-образовательную среду филиала,</p>	<p>353919, Краснодарский край, г. Новороссийск, ул. Мысхакское шоссе, дом № 75, аудитория № 407 35,5 кв.м., этаж 4, помещение 407</p>

<p>мультимедийным проектором и экраном, веб-камерой, графическим планшетом, программным пакетом Microsoft Windows 7 Профессиональная, Microsoft Office Стандартный 2007 (академическая лицензия № 49190957 от 20.10.2011); Dr. Web Security Space 12 - сублицензионный договор 490 от 10.08.2021; браузеры Google Chrome, Internet Explorer, Zoom, Sumatra PDF, 7Zip – свободно распространяемое ПО согласно условиям лицензионного соглашения</p>	
<p>Читальный зал библиотеки № 410 для самостоятельной работы с выходом в сеть Интернет. Оснащен специализированной мебелью, кондиционером, персональными компьютерами с выходом в Интернет и обеспечением доступа в электронную информационно-образовательную среду филиала, веб-камерой, графическим планшетом, программным пакетом Microsoft Windows 7 Профессиональная, Microsoft Office Стандартный 2007 (академическая лицензия № 49190957 от 20.10.2011); Dr. Web Security Space 12 - сублицензионный договор 490 от 10.08.2021; браузеры Google Chrome, Internet Explorer, Zoom, Sumatra PDF, 7Zip – свободно распространяемое ПО согласно условиям лицензионного соглашения</p>	<p>353919, Краснодарский край, г. Новороссийск, ул. Мысхакское шоссе, дом № 75, аудитория № 410 35:4 кв.м., этаж 4, помещение 410</p>

3.2. Доступная среда

В НФ БГТУ им. В.Г. Шухова при создании безбарьерной среды учитываются потребности следующих категорий инвалидов и лиц с ограниченными возможностями здоровья:

- с нарушениями зрения;
- с нарушениями слуха;
- с ограничением двигательных функций.

В образовательной организации обеспечен беспрепятственный доступ в здание инвалидам и лицам с ограниченными возможностями здоровья.

Для лиц с нарушением работы опорно-двигательного аппарата обеспечен доступ для обучения в аудиториях, расположенных на первом этаже, также имеется возможность доступа и к другим аудиториям.

Для лиц с нарушением зрения, слуха имеется аудитория, обеспеченная стационарными техническими средствами.

В сети «Интернет» есть версия официального сайта учебной организации для слабовидящих.

3.3. Информационное обеспечение обучения

Основные источники

1. Трофимов, В. В. Основы алгоритмизации и программирования: учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская; под редакцией В. В. Трофимова. — 4-е изд. — Москва: Издательство Юрайт, 2023. — 119 с. — (Профессиональное образование). — Текст: непосредственный. ISBN 978-5-534-17498-4 URL: <https://urait.ru/viewer/osnovy-algoritmizacii-i-programmirovaniya-533200#page/1> Доступ по подписке

Дополнительная литература:

2. Федоров, Д. Ю. Программирование на языке высокого уровня Python: учебное пособие для среднего профессионального образования / Д. Ю. Федоров. — 5-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2023. — 227 с. — (Профессиональное образование). — Текст: непосредственный. ISBN 978-5-534-17319-2 Доступ по подписке URL: <https://urait.ru/viewer/programmirovanie-na-yazyke-vysokogo-urovnya-python-532858#page/1> Доступ по подписке

Электронные библиотеки

1. Научная электронная библиотека e-LIBRARY.RU (<http://elibrary.ru>).
2. Российская государственная библиотека (РГБ) (www.rsl.ru)
3. Электронная библиотека БГТУ им. В.Г. Шухова на базе ПО «БиблиоТех» (<https://elib/bstu.ru/>)
4. Электронная библиотечная система издательства «Лань» (<http://e/lanbook.com/>)

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Контроль и оценка результатов освоения дисциплины осуществляется преподавателем в процессе проведения практических занятий, тестирования.

Проверяемые компетенции (код): ОК 1, ОК 2, ОК 9, ПК 2.5	
Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля и оценки результатов обучения
<p>Уметь:</p> <ul style="list-style-type: none"> - Разрабатывать алгоритмы для конкретных задач. -Использовать программы для графического отображения алгоритмов. -Определять сложность работы алгоритмов. -Работать в среде программирования. -Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. -Оформлять код программы в соответствии со стандартом кодирования. -Выполнять проверку, отладку кода программы. 	<p>Устный опрос на знание терминологии по теме; Наблюдение за выполнением практического задания. Оценка выполнения практического задания Тестирование (6, 12 недели семестра) Дифференцированный зачет, экзамен</p>
<p>Знать:</p> <ul style="list-style-type: none"> -Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. -Эволюцию языков программирования, их классификацию, понятие системы программирования. -Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти. 	<p>Устный опрос на знание терминологии по теме; Наблюдение за выполнением практического задания. Оценка выполнения практического задания Тестирование (6, 12 недели семестра) Дифференцированный зачет, экзамен</p>
	<p>Итоговый контроль в форме: Диф.зачета Экзамена</p>

4.1 Показатели оценки образовательных результатов

Проверяемые компетенции (код): ОК 1, ОК 2, ОК 9, ПК 2.5	
Образовательные результаты (знания, умения)	Показатели оценки результата
- разрабатывать алгоритмы для конкретных задач;	Решение задач с помощью конкретных алгоритмов.
- использовать программы для графического отображения алгоритмов;	Отображение графического представления алгоритмов.

- определять сложность работы алгоритмов;	Определение сложности работы алгоритмов
- работать в среде программирования.	Выбор среды программирования.
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;	Разработка построенных алгоритмов в виде программ на конкретном языке программирования.
- оформлять код программы в соответствии со стандартом кодирования;	Оформление кода программы в соответствии со стандартом кодирования
- выполнять проверку, отладку кода программы.	Отладка кода программы.
- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;	Формулирование понятия алгоритмизации, свойств алгоритмов, общих принципов построения алгоритмов, основных алгоритмических конструкций.
- эволюцию языков программирования, их классификацию, понятие системы программирования;	Описание эволюции языков программирования, их классификацию, понятие системы программирования.
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;	Описание основных элементов языка, структуры программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти
- подпрограммы, составление библиотек подпрограмм;	Выбор и разработка подпрограмм, составление библиотек подпрограмм
- объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.	Описание объектно-ориентированной модели программирования, основных принципов объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

4.2 Перечень вопросов для устного опроса

3 СЕМЕСТР

1. Что такое алгоритм?
2. Что такое язык программирования?
3. Приведите примеры областей применения языков программирования.
4. Что относится к стандартам языков программирования.
5. Что такое среда программирования?
6. Для чего нужны компиляторы и интерпретаторы?
7. Назовите жизненные циклы программы.
8. Что такое программа?
9. Перечислите основные этапы решения задач на компьютере.

10. Перечислите основные фигуры блок-схемы алгоритма
11. Перечислите основные типы данных языка Python.
12. Что относится к простым типам данных?
13. Какие операции и выражения можно применять к числовому типу данных?
14. Приоритеты в вычислениях.
15. Назовите операторы ввода и вывода данных.
16. Что делает оператор присваивания?
17. Что такое составной оператор? Приведите примеры
18. Какие виды условного оператора вы знаете?
19. Что такое цикл?
20. В чем разница циклов For и While?
21. Назовите правило инициализации строчных элементов.
22. Перечислите стандартные процедуры и функции для работы со строками.
23. Способы инициализации списка
24. Основные функции и методы работы со списками

Шкала оценки устного ответа

Критерии	Качественная оценка образовательных результатов	
	балл (отметка)	вербальный аналог
Дан развернутый ответ на поставленный вопрос	5	отлично
В ответе допущен один-два недочета и (или) одна ошибка	4	хорошо
В ответе допущено несколько недочётов и две ошибки	3	удовлетворительно
В ответе допущено несколько недочетов и более двух ошибок	2	неудовлетворительно

4.2.1. Перечень заданий для контроля умений образовательных результатов

1. Нарисуйте алгоритм и напишите программу, запрашивающую стороны прямоугольника А и В и находящую его периметр и площадь
2. Нарисуйте алгоритм и напишите программу, запрашивающую *имя пользователя и имя его друга*, а также выводящую на экран строки по данному образцу:

Как Вас зовут? Иван
Как зовут Вашего друга? Николай
Иван и Олег – друзья!
3. Нарисуйте алгоритм и напишите программу, запрашивающую ввод двух целых чисел и нахождение их суммы, произведения, разности и частного по образцу:

Введите А:

Введите В:
Сумма чисел А и В равна ...
Произведение чисел А и В равно ...
Разность чисел А и В равна ...
Частное чисел А и В равно ...

4. Нарисуйте алгоритм и напишите программу, запрашивающую ввод расстояния до пункта назначения (в км), скорость транспорта (км/ч) и рассчитывающую время поездки по образцу:

Введите расстояние до пункта назначения (км):
Введите скорость транспорта (км/ч):
Время поездки составит: ... ч.

5. Нарисуйте алгоритм и напишите программу, запрашивающую ввод значения температуры в Цельсиях и переводящая ее в Фаренгейты ($F=C*9/5+32$) с выводом результата на экран по образцу:

Введите T в Цельсиях(C):
T в Фаренгейтах составит:

6. Нарисуйте алгоритм и напишите программу, запрашивающую количество студентов S и количество ручек G и определяющую сколько ручек (поровну) достанется каждому студенту, и сколько останется неподделённых.

Введите кол-во студентов:
Введите кол-во ручек:
У каждого студента по ... ручек и неподделенных.

7. Обозначьте порядок вычисления выражения по операциям:

$$11 * 2 ** 2 - 12 / 4 + 7.$$

Какое число получится в итоге?

8. Напишите код перевода следующих переменных в целые числа:

3.93; -11.328
- в дробные числа:
8; -17

9. Напишите программу, которая считывает два натуральных числа a и b и выводит наибольшее значение из них. Используйте функцию max:

102 и 202
05 и 06
Анна и Инна

10. Нарисуйте алгоритм и напишите программу, запрашивающую четырёхзначное число и подсчитывающее сумму его цифр.

11. С начала некоторых суток прошло n минут. Определите, сколько часов и минут будут показывать электронные часы в этот момент. На вход программе подается целое неотрицательное число n. Программа должна вывести два числа: количество часов (от 0 до 23) и количество минут (от 0 до 59).

12. Существует список данных: Animals с названиями животных (мышь, лошадь, курица,

сова, змея, мартышка). Выведите на экран 0-элемент этого списка

13. Создайте список из 7 чисел. Используя функцию MAX и MIN выведите на экран максимальное и минимальное числа из этого списка по образцу:

Максимальное число списка: _____

Минимальное число списка: _____

14. Найти скорость движения автомобиля, если путь S вводится программой случайно, а время в пути t – пользователем с клавиатуры по образцу:

Введите время в пути t:

Путь S=

Скорость автомобиля V=

15. Какими будут значения переменных m и n после выполнения группы операторов?

M=25

N=M+1

M=M-25

Print('N= ',N)

Print('M= ',M)

16. Какими будут значения переменных m и n после выполнения группы операторов?

M=20

N=10

M=M/N

*N=M*N*

N=N+30

Print('N= ',N)

17. Какими будут значения переменных m и n после выполнения группы операторов?

M=30

N=2

N=M/2

M=N

M=M+N

M=M

Print('M= ',M)

18. Разработайте алгоритм и программу, в которой вычисляются площадь и объем сферы:

$S_{\text{сф}}=4\pi R^2$; ; - pi

19.

20. Разработайте программу, в которой запрашивается слово из 5 букв. Программа меняет местами первые две и последние две буквы, например ГРОЗА - ЗАОГР

21. Разработайте программу, запрашивающую данные пользователя и выводящую на экран некий шаблон документа по образцу:

Введите город:

Введите улицу:

Введите номер дома:

Введите номер квартиры:

Введите фамилию:

Введите имя:
Введите отчество:

Новороссийск
Ул. Советов, д.56, кв.15
Иванов И.И.

Уважаемый, Иван Иванович!

22. Каким будет значение переменной s после выполнения группы операторов?

```
w=3
p=5
j=3.5
if (j<p) and (j>w):
    j=j+0.5
    j=j+12
else:
    j=11
print("Значение j=", j)
```

23. Каким будет значение переменной s после выполнения группы операторов?

```
n=2.5
f=0.5
d=4
s=0
if n<f:
    s=12
if f>=n:
    s=28
if d==4:
    s=39
print("s=", s)
```

24. Каким будет значение переменной j после выполнения группы операторов?

```
w=3
p=5
j=3.5
if (j<p) and (j>w):
    j=j+0.5
    j=j+10
else:
    j=11
print("j=", j)
```

25. Каким будет значение переменной j после выполнения группы операторов?

```
j=6
k=6
if j>k:
    j=j+2
    j=j+3
```

```
else:
    j=k-3
    j=j+4
    print("Значение j=", j)
```

26. Каким будет значение переменной j после выполнения группы операторов?

```
w=3
p=5
j=3.5
if (j<p) and (j>w):
    j=j+0.5
    j=j+12
else:
    j=11
print("Значение j=", j)
```

27. Для посещения аттракционов иногда устанавливаются некоторые ограничения на возраст, рост и т.д. Нарисуйте алгоритм и напишите программу проверяющие – допускается человек на аттракцион или нет по образцу:

Укажите свой возраст:

Укажите свой рост (см):

Извините, Вы не допущены (:

или

Добро пожаловать! :)

28. Чтобы не так легко было взломать сервер, логин пользователя должен быть не меньше 4-х символов, а пароль не менее 5 и не более 9 символов. Реализуйте такую проверку имени и пароля пользователя.

Введите логин:

Введите пароль

Введите правильный логин или пароль! (если введено неправильно)

Вход выполнен! (если введено правильно)

29. Нарисуйте алгоритм вводящий два числа – делимое и делитель. Если делитель равен нулю, выдать сообщение, что на ноль делить нельзя, в противном случае, найти: результат деления, целую часть от деления и остаток от деления по образцу:

Введите два числа:

Результат деления:

Целая часть:

Остаток от деления:

30. Нарисуйте следующий алгоритм: планируем свой завтрашний день. Если будет хорошая погода, то проведем день в лесу. Если же погода будет плохая, то сначала займемся

уборкой квартиры, а во второй половине дня ходим в зоопарк.

31. Нарисуйте следующий алгоритм и напишите программу нахождения корней квадратного уравнения $ax^2 + bx + c = 0$

32. Сделайте трассировку следующего кода:

```
D=4
R=15
for i in range (D+1, R,1):
    B=5
```

33. Сделайте трассировку следующего кода:

```
r=50
s=0
for i in range(5,0,-1):
    s=1
    R=r-s
    Print("r=",R)
```

34. Сделайте трассировку следующего кода:

```
a=7
d=5
y=0
for i in range(1,4,1):
    y=d
    y=a+2
    Print("y=",y)
```

35. Сделайте трассировку следующего кода:

```
a=17.0
d=a
for i in range(3):
    print("i=",i)
    if a!=d:
        a=a+1
    else:
        y=a
    Print ("y=",y)
```

36. Сделайте трассировку следующего кода:

```
A=1
Y=1
for i in range(2,6,1):
    A=A+10
    Y=A+10
    Print("y=",y)
```

37. Сделайте трассировку следующего кода:

```

a=5
d=5
y=0
for i in range(7,1,-1):
    a=d
    y=a+10
Print ("y=",y)

```

38. Сделайте трассировку следующего кода:

```

a=5
d=5
y=0
for i in range(2,6,1):
    y=d+10
Y=a+10
Y=a*d
Print("y=",y)

```

39. Определите количество повторов цикла:

```

n=2
s=13
y=0
for i in range(s+2,n+3,-1):
    y=y+1

```

40. Нарисуйте алгоритм и напишите программу, последовательно запрашиваемую ввод шести чисел. Вывести на экран каких больше: положительных или отрицательных

41. Нарисуйте алгоритм и напишите программу, последовательно запрашиваемую ввод пяти чисел. Вывести на экран минимальное из всех

42. Нарисуйте алгоритм и напишите программу, последовательно запрашиваемую ввод N чисел. Вывести на экран количество отрицательных чисел и минимальное из них

43. Нарисуйте алгоритм и напишите программу, последовательно запрашиваемую ввод N чисел. Вывести на экран количество положительных чисел и максимальное из них

44. Определите, какое значение находится в ячейке Y после выполнения группы операторов:

```

K=0
Y=3
While K<Y:
    Y=Y+1
    K=K+3
Print('y=',Y)

```

45. Каким будет значение переменной s после выполнения группы операторов:

```

s=7

```

```
i=1
While i*I ==2:
s=s+1/i
i=i+1
```

46. Определите, какое значение находится в ячейке a после выполнения группы операторов:

```
a=2
d=1
While a+d<=7:
a=a+1
d=d+1
a=a*d
```

47. Каким будет значение переменной z после выполнения группы операторов:

```
a=1
z=4
While a<=3:
a=a+1
z=a+1
Print('z=',z)
```

48. Пользователь имеет какую-то сумму наличности на балансе. Он может совершать покупки, пока баланс не будет меньше 5000 рублей. Написать программу, выводящую предупреждение: *Извините, Вы превысили сумму покупки*

49. Программа, должна считать сумму чисел до тех пор, пока мы не ввели 0, но что-то пошло не так. Решите эту проблему:

```
number = int(input('Введите число: '))
while number <= 0:
    summ= number + summ
    number = int(input('Введите число: '))
print(summ)
```

50. Напишите программу, которая выводит на экран вот такую последовательность чисел (каждое число в новой строчке): 7 14 21 28 35 42 49 56 63 70 77 84 91 98

Найдите закономерность и выведите значения используя цикл.

51. Определите, какое значение находится в ячейке n после выполнения группы операторов:

```
n=2
x=1
While x<=4:
x=x+1
n=n+x
```

52. Каким будет значение переменной a после выполнения группы операторов:

```
a=1
z=1
```

```
While a<=3:
a=a+1
a=a+z
a=a+10
```

53. Определите, какое значение находится в ячейке s после выполнения группы операторов:

```
a=20
d=5
while a-10>=10:
a=a-1
d=d+1
S= a+d
```

54. Каким будет значение переменной a после выполнения группы операторов:

```
a=1
z=1
While z<4:
z=z+z
z=a+1
a=z
```

55. Нарисуйте алгоритм и напишите программу имитирующую работу с приложением:

- спрашивает у пользователя «Продолжаем работать? 1/0:» до тех пор, пока пользователь не введёт 0
- выводится сообщение: «Приложение закрывается...»
- в конце программы выводится сообщение: «Работа завершена»

56. Нарисуйте алгоритм и напишите программу, которая выводит на экран вот такую последовательность чисел (каждое число в новой строчке):

7 14 21 28 35 42 49 56 63 70 77 84 91 98.

Найдите закономерность и выведите значения используя бесконечный цикл и прерывание break.

57. На экран выводятся числа с 10 до 0 и сообщение «Время вышло!». Необходимо исправить ошибку и применить бесконечный цикл и прерывание Break:

```
count = 0
while count <= 10
if count == 0:
print('Время вышло!')
else:
print(count)
count = count - 1
```

58. Нарисовать алгоритм и написать программу ввода последовательности чисел. Известно, что последний элемент равен 5. Найти количество положительных чисел, используя бесконечный цикл и прерывание break

59. Реализуйте следующую программу, используя бесконечный цикл и прерывание break:

Некая программа блокирует работу, выдавая сообщение о блокировке

компьютера, например:

Ваш компьютер заблокирован! Введите правильный код!

Как только вводится правильный код – выдается сообщение : «Код верный, завершаю работу ...»

60. Нарисовать алгоритм и написать следующую программу: с помощью приложения проводится розыгрыш призов среди подписчиков с определенными номерами (6732, 115, 389, 225, 661, 2905, 117, 484). Выигрышными будут считаться те номера, которые кратны 5 и состоят из трех цифр. Вывести список выигравших номеров и их количество по образцу:

*Билет № ... - Выиграл!
Билет № ... - Выиграл!
Кол-во выигрышных билетов - ...*

61. Нарисовать алгоритм и написать следующую программу: приложение запрашивает ввод двух положительных чисел, например А и В, причем первое меньше второго. А затем находит среднее арифметическое чисел лежащих в диапазоне от А до В, кратных трем.

62. Сделайте трассировку следующего кода:

```
l = 0
r = 100
middle = 0
print('Загадайте число! ')
middle = (L+R)//2
while l <= r:
    print('Число ',middle,' больше(2) искомого / меньше (3) / или равно ему (1)?')
    quest = int(input())
    if quest == 1:
        print('искомое число', middle)
        break
    elif quest == 3:
        l = middle
        middle = (r + l)//2
    if middle + 1 == l:
        print('искомое число', middle)
        break
    elif quest == 2:
        r = middle
        middle = (r + l)//2
    if middle - 1 == r:
        print('искомое число', middle)
        break
```

63. Сделайте трассировку следующего кода:

```
count = 0
while True:
    number = int(input('Введите число: '))
    count += 1
    if number == 7:
```

```

        print('Вы угадали! Число попыток: ',count)
        break
    elif number < 7:
        print('Число меньше чем нужно. Попробуйте еще раз!')
    else:
        print('Число больше чем нужно. Попробуйте еще раз!')

```

64. Сделайте трассировку следующего кода:

```

summ = 0
a = [3,7, 5,9,3,4]
for i in range(6):
    print(a[i])

```

65. Нарисовать алгоритм и написать программу нахождения суммы элементов списка кратным трем

66. Сделайте трассировку следующего кода:

```

animals = ['cat', 'dog', 'bat']
i = 0
while i < len(animals):
    print(animals[i])
    i += 1

```

67. Что выведется на экран в результате выполнения этих кодов?:

- 1)

```
a = [1, 2, 3]
a.append(4)
print(a)
```
- 2)

```
a = [1, 2, 3]
a.extend([4, 5])
print(a)
```
- 3)

```
a = ['cat', 'dog', 'bat']
a.extend('mouse')
print(a)
```
- 4)

```
a = [1, 2, 3]
a.insert(0, 4)
print(a)
```

68. Реализовать программу, формирующую список $x = [-1.5, 0.1, 12, 0, -2.2, 0.5, -1, 0.3]$, вычисляющую и выводящую сумму и кол-во элементов списка, принадлежащих отрезку $[0,1]$

69. Реализовать программу, формирующую список $x = [-1.5, 0.1, 12, 0, -2.2, 0.5, -1, 0.3]$, а затем переписывающую элементы списка, принадлежащие отрезку $[-1,1]$ в список y и выводящую её на экран

70. Реализовать программу, формирующую список $x = [2.5, 4.9, 10.2, -7.12, 3.1, -2, 6]$, а затем формирующую новый список z , элементами которого будут являться положительные элементы списка x , и упорядочите их по возрастанию

Критерии оценки образовательных результатов

Шкала оценки устных ответов

Критерии	Качественная оценка образовательных результатов.	
	балл (отметка)	вербальный аналог
Поставленная задача (проблема) выполнена полностью. В решении прослеживается логика, учтены структура и стандарты в написании программы	5	отлично
Поставленная задача (проблема) выполнена не до конца. В решении прослеживается логика, но не учтены структура и стандарты в написании программы. Ответы на вопросы сигнализируют о наличии проблемы в понимании определенной темы.	4	хорошо
Поставленная задача (проблема) выполнена не до конца. В решении прослеживается логическое недопонимание структуры программы или этапов ее решения. Стандарты в написании программы учтены. Решение задания при помощи преподавателя.	3	удовлетворительно
Поставленная задача (проблема) не выполнена. В решении прослеживается логическое непонимание структуры программы и этапов ее решения. Стандарты в написании программы не учтены. Подсказки преподавателя для решения задачи не помогли	2	неудовлетворительно

4.3 Тест для промежуточной аттестации (6 неделя)

Номер задания	Содержание вопроса/задания
1.	Что выведет данный код? A = list(-5) print(A)
2.	Что выведет данный код? s= int('-2') print(s)
3.	Что выведет данный код? s= int('-2') print(len(s))
4.	Что выведет данный код? s= str(10.0)

	print(s)
5.	Что выведет данный код? d = bool('True') print(d)
6.	Что выведет данный код? d= "2" - 1 print(d)
7.	Что выведет данный код? d = dict(("01","py")) print(len(d))
8.	Что выведет данный код? x = '7' y = '3' z = x + y print(z)
9.	Что выведет данный код? x = '5' y = '2' z = x*int(y) print(z)
10.	Как в Пайтоне создать пустой список? а) li = [], б) li = None, в) li = 0, г) li = list().
11.	Выберите верные утверждения о списках. а) неизменяемые, б) упорядоченные, в) последовательности, г) изменяемые
12.	Найдите соответствия: а) dict, б) list, в) tuple 1- кортеж, 2- словарь, 3 – список

Ключ ответов

№ вопроса	Верный ответ
1.	Ответ: ошибку
2.	Ответ: -2
3.	Ответ: ошибку
4.	Ответ:10
5.	Ответ: ошибку
6.	Ответ: 10.0
7.	Ответ: 2
8.	Ответ:73
9.	Ответ:55

10.	Ответ:а,г
11.	Ответ: в,г
12.	Ответ: a-2, b-3, b - 1

4.4 Тест для промежуточной аттестации (12 неделя)

Номер задания	Содержание вопроса/задания
1.	<p>Запись</p> <pre>if <условие>: оператор</pre> <p>называется ...</p> <p>а) полной формой условного оператора б) неполной формой условного оператора в) составным условием г) простым условием</p>
2.	<p>Составные условия строятся из простых с помощью:</p> <p>а) логических операций and, or, not б) арифметических операций +, -, * в) с помощью операций and, or, not, +, -, *</p>
3.	<p>Каким будет значение переменной s после выполнения группы операторов?</p> <pre>n=2.5 f=0.5 d=4 s=0 if n<f: s=12 if f>=n: s=28 if d==4: s=39 print("s=", s)</pre>
4.	<p>Сделайте трассировку следующего программного кода. Каким будет значение переменной y после выполнения группы операторов?</p> <pre>a=5 d=5 y=0 for i in range(2,6,1): y=d+10 Y=a+10 Y=a*d Print("y=",y)</pre>
5.	<p>Каким будет значение переменной j после выполнения группы операторов?:</p> <pre>j=3 k=15 m=20 if j<=k: if m>k: j=k%2 j=j%3 else: j=10 print("Значение j=", j)</pre>
6.	Составьте алгоритм и напишите программный код, реализующий выполнение

	следующей задачи: последовательно вводятся шесть чисел. Вывести на экран каких больше: положительных или отрицательных.
--	---

Ключ ответов

№ вопроса	Верный ответ
1.	Ответ: ошибку
2.	Ответ: б,в
3.	Ответ: 39
4.	Ответ: 15
5.	Ответ: 1

Шкала оценивания и перевода баллов в оценки

Процент выполнения	Количество баллов	Оценка
100% - 90%	105-95	«5»
89% - 80%	94-84	«4»
79% - 70%	83-73	«3»
69% и меньше	72 и меньше	«2»

ВОПРОСЫ НА ЭКЗАМЕН

1. Алгоритм.
2. Способы описания алгоритма.
3. Линейный алгоритм
4. Условный алгоритм
5. Циклический алгоритм
6. Язык программирования. Разновидности
7. Компилятор.
8. Этапы компиляции
9. Интерпретаторы
10. Переменные. Правила создания
11. Типы переменных.
12. Математические операторы. Приоритеты в вычислениях
13. Операторы присваивания
14. Ввод и вывод информации
15. Модули. Подключение
16. Строки. Работа со строками
17. Условный оператор. Виды
18. Циклы. Виды

19. Бесконечный цикл и прерывание
20. Списки. Работа со списками
21. Функции и методы
22. Бинарный поиск
23. Использование комментариев

КРИТЕРИИ ОЦЕНИВАНИЯ ТЕОРЕТИЧЕСКОЙ ЧАСТИ БИЛЕТА

Оценка	Критерии оценки ответа студента
«Отлично»	<p>Обстоятельно и с достаточной полнотой излагает материал вопросов. Даёт ответ на вопрос в определенной логической последовательности. Даёт правильные формулировки, точные определения понятий и терминов.</p> <p>Демонстрирует полное понимание материала, даёт полный и аргументированный ответ на вопрос, приводит необходимые примеры (не только рассмотренные на занятиях, но и подобранные самостоятельно).</p> <p>Свободно владеет речью (показывает связанность и последовательность в изложении).</p>
«Хорошо»	<p>Даёт ответ, удовлетворяющий тем же требованиям, что и для оценки «отлично», но допускает единичные ошибки, неточности, которые сам же исправляет после замечаний преподавателя.</p>
«Удовлетворительно»	<p>Обнаруживает знание и понимание основных положений, но:</p> <ul style="list-style-type: none"> - допускает неточности в формулировке определений, терминов; - излагает материал недостаточно связно и последовательно; - на вопросы экзаменаторов отвечает некорректно.
«Неудовлетворительно»	<p>Обнаруживает непонимание основного содержания учебного материала. Допускает в формулировке определений ошибки, искажающие их смысл.</p> <p>Допускает существенные ошибки, которые не может исправить при наводящих вопросах преподавателя или ответ отсутствует.</p> <p>Беспорядочно и неуверенно излагает материал.</p> <p>Сопровождает изложение частыми заминками и перерывами.</p>

КРИТЕРИИ ОЦЕНИВАНИЯ ПРАКТИЧЕСКОЙ ЧАСТИ БИЛЕТА

Оценка	Критерии
«Отлично»	Показал полное знание технологии выполнения задания. Продemonстрировал умение применять теоретические знания/правила выполнения/технологию при выполнении задания. Уверенно выполнил действия согласно условию задания.
«Хорошо»	Задание в целом выполнил, но допустил неточности. Показал знание технологии/алгоритма выполнения задания, но недостаточно уверенно применил их на практике. Выполнил норматив на положительную оценку.
«Удовлетворительно»	Показал знание общих положений, задание выполнил с ошибками. Задание выполнил на положительную оценку, но превысил время, отведенное на выполнение задания.
«Неудовлетворительно»	Не выполнил задание. Не продемонстрировал умения самостоятельного выполнения задания. Не знает технологию/алгоритм выполнения задания. Не выполнил норматив на положительную оценку.

Оценка	Критерии
«Отлично»	Показал полное знание технологии выполнения задания. Продemonстрировал умение применять теоретические знания/правила выполнения/технологию при выполнении задания. Уверенно выполнил действия согласно условию задания.
«Хорошо»	Задание в целом выполнил, но допустил неточности. Показал знание технологии/алгоритма выполнения задания, но недостаточно уверенно применил их на практике. Выполнил норматив на положительную оценку.
«Удовлетворительно»	Показал знание общих положений, задание выполнил с ошибками. Задание выполнил на положительную оценку, но превысил время, отведенное на выполнение задания.
«Неудовлетворительно»	Не выполнил задание. Не продемонстрировал умения самостоятельного выполнения задания. Не знает технологию/алгоритм выполнения задания. Не выполнил норматив на положительную оценку.

4 СЕМЕСТР

4.5 Перечень вопросов для устного опроса

1. Что такое мощность множества?
2. Какой аналог есть у списков?
3. Как удалить элемент множества, не вызывая при этом ошибку, если элемента в списке нет?
4. В чем суть объединения множеств? Приведите два способа и поясните разницу
5. Что собой представляет супермножество?
6. Почему поиск в множестве быстрее, чем в списке?
7. Что такое функция и для чего его используют?
8. Какие бывают в функции?
9. В чем отличие между параметрами и аргументами функции?
10. С помощью какого ключевого слова можно создать функцию?
11. Что такое указатель в питоне?.
12. Что такое ООП?
13. Для чего используют ООП?
14. Назовите основные принципы ООП
15. Что такое объект, класс?
16. В чем разница между классом и объектом?
17. Что такое наследование? Виды наследования?
18. Что суперкласс, подкласс, полиморфизм?
19. Что такое метод объекта?
20. Укажите несуществующие свойства компонента Edit
21. Системы программирования, позволяющие создавать программы с оконным интерфейсом называют ____ приложениями (оконными)
22. Какое ключевое слово используется в Python для определения классов? а) class, б) cls, в) object, г) obj
23. Какой стиль именования классов рекомендуется в руководстве PEP 8 по оформлению кода на Python? а) camelCase, б) CapWords, в) snake_case, г) ALL_CAPS.
24. Какой оператор используется для доступа к атрибуту объекта или класса в Python? а) -, б) ::, в) ., г) *.

Шкала оценки устного ответа

Критерии	Качественная оценка образовательных результатов	
	балл (отметка)	вербальный аналог
Дан развернутый ответ на поставленный вопрос	5	отлично
В ответе допущен один-два недочета и (или) одна ошибка	4	хорошо
В ответе допущено несколько недочётов и две ошибки	3	удовлетворительн о
В ответе допущено несколько недочетов и более двух ошибок	2	неудовлетворител ьно

4.6 Тест для промежуточной аттестации (6 неделя)

Номер задания	Содержание вопроса/задания
1.	<p>Запись</p> <pre style="text-align: center;">if <условие>: оператор</pre> <p>называется ...</p> <p>а) полной формой условного оператора б) неполной формой условного оператора в) составным условием г) простым условием</p>
2.	<p>Укажите основные типы событий:</p> <p>а) производимые мышью б) нажатие клавиш на клавиатуре в) события, возникающие в результате изменения виджетов.</p>
3.	<p>Что выведет данный код?</p> <pre>class Nums: def __init__(self, num): self.num = num print(f'1000 - {self.num}') Nums('7')</pre> <p>а) 993 б) 1000-7 в) 1000 – 7 г) Ошибку</p>
4.	<p>Что выведет данный код?</p> <pre>class Spam: val = 1 def __init__(self, n): val = 5 obj = Spam(5) print(obj.val)</pre> <p>а) 6 б) 5 в) 1 г) 0</p>
5.	<p>Что выведет данный код?</p> <pre>class A: def __init__(self): self.a = 0 def change(self, n): a = n obj = A() obj.change(2) print(obj.a)</pre> <p>а) 0 б) 2 в) 1</p>
6.	<p>Что выведет данный код?</p> <pre>class Person(): def __init__(self, name): self.name = name</pre>

	<pre>def get_name(self): return name people = Person("Alex") print(people.get_name())</pre> <p>a) Alex б) None в) Ошибку г) Ничего</p>
7.	<p>Что выведет данный код?</p> <pre>class Apple: def func1(b): return a + b def func(): global a a = 1 Apple.func1(10) print(Apple.func())</pre> <p>a) 10 б) None в) 2 г) Ничего</p>
8.	<p>Что выведет данный код?</p> <pre>class Parent(): x = 1 y = 2 class Child(Parent): x = 111 y = 222 def mix(self): return Parent.y c = Child() print(c.mix())</pre> <p>a) 222 б) 2 в) 111 г) 1</p>
9.	<p>Что выведет данный код?</p> <pre>class Egg: __eggs = 8 def get_eggs(self): print(self.__eggs) egg = Egg() egg.get_eggs()</pre> <p>a) 8 б) 0 в) Ошибку г) None (Ответ: a)</p>
10.	<p>Что выведет данный код?</p> <pre>class Stranger: def __init__(self,x): self.x = x</pre>

	<pre>def __str__(self): return str(self.x) def __call__(self, n=5): return self.x * n @classmethod def z(self, n=7): return self(n) s = Stranger(3) print(s.z())</pre> <p>а) 5 б) 3 в) 7 г) Ошибку</p>
11.	<p>Что выведет данный код?</p> <pre>class A: pass print(A.__class__)</pre> <p>а) class 'type' б) type 'class' в) Ошибку г) Ничего</p>
12.	<p>Что выведет данный код?</p> <pre>class A: def __init__(self): self.a = 1 self.__c = 3 a = A() try: print(a.a) print(a.__c) except Exception: print(5)</pre> <p>а) 1 3 б) 1 2 в) 1 5 г) Ошибку</p>

Ключ ответов

№ вопроса	Верный ответ
1.	Ответ: б
2.	Ответ: а, б, в
3.	Ответ: в
4.	Ответ: в
5.	Ответ: а
6.	Ответ: в
7.	Ответ: б
8.	Ответ: б
9.	Ответ: а
10.	Ответ: в

11.	Ответ: а
12.	Ответ: в

4.7 Тест для промежуточной аттестации (12 неделя)

Номер задания	Содержание вопроса/задания
1.	<p>Что такое функция?</p> <p>а) Объект, принимающий аргументы и возвращающий значение б) Структура, определяющая поведение объекта в) Любой код в Python г) Код, заключенный в круглые скобки</p>
2.	<p>Какое ключевое слово используется для создания функции?</p> <p>а) fun б) function в) void г) def</p>
3.	<p>Сколько параметров может принимать функция?</p> <p>а) Нисколько, функция не принимает значения, только возвращает б) 1 в) 2 г) Бесконечно много</p>
4.	<p>Какое ключевое слово используется для возврата значения из функции?</p> <p>а) get б) post в) return г) answer</p>
5.	<p>Что выведет этот код?</p> <pre>def get_summ(a,b): pass print(get_summ(2,4))</pre> <p>а) 4 б) 2 в) 6 г) None</p>
6.	<p>Каким ключевым словом обозначается анонимная функция?</p> <p>а) lambda б) alpha в) anonym г) Таких функций не существует</p>
7.	<p>Что выведет этот код?</p> <pre>def get_summ(a=2,b=3): print(a+b) get_summ(4)</pre> <p>а) 5 б) 4 в) 7 г) 6</p>

8.	<p>Для чего нужно ключевое слово global?</p> <p>а) Чтобы сделать переменную доступной из любой точки мира б) Чтобы указать, что эта переменная имеет самое важное значение в программе в) Такого ключевого слова нет, мы достали его из языка C++ г) Чтобы переменную можно было изменять за пределами текущей области видимости</p>
9.	<p>Что такое рекурсивная функция?</p> <p>а) Функция, которая курсирует между модулями б) Функция которая возвращает саму себя в) Функция, которая выполняется с последней строки по первую г) Функция, написанная курсивом</p>
10.	<p>В каком случае правильно создана анонимная функция?</p> <p>а) lambda x+1 : x б) x + 1 = lambda x в) lambda x : x + 1 г) def lambda():</p>

Ключ ответов

№ вопроса	Верный ответ
1.	Ответ: а
2.	Ответ: г
3.	Ответ: г
4.	Ответ: в
5.	Ответ: г
6.	Ответ: а
7.	Ответ: в
8.	Ответ: г
9.	Ответ: б
10.	Ответ: в

Шкала оценивания и перевода баллов в оценки

Процент выполнения	Количество баллов	Оценка
100% - 90%	105-95	«5»
89% - 80%	94-84	«4»
79% - 70%	83-73	«3»
69% и меньше	72 и меньше	«2»

4.8 Вопросы к дифференцированному зачету (теоретическая часть)

1. Дать общую характеристику языка программирования Python
2. Охарактеризовать процессы тестирования и отладки программ.
3. Перечислить и охарактеризовать свойства алгоритма.
4. Дать определение понятия «язык программирования». Перечислить критерии классификации языков программирования и охарактеризовать их.
5. Перечислить и оценить типовые алгоритмы для работы с линейными массивами.
6. Перечислить и охарактеризовать основные объекты языка Python.
7. Описать задачу сортировки списка. Оценить алгоритм сортировки линейного массива методом "пузырька"
8. Рассказать, что такое среда программирования, дать характеристику компонентам, ее составляющим.
9. Дать определение и краткую характеристику синтаксических и логических ошибок, привести примеры.
10. Оценить использование динамической памяти: выделение и освобождение памяти при работе со списками/словарями. Перечислить особенности обработки словарей при работе с динамической памятью.
11. Дать характеристику приоритетам операций в Python.
12. Перечислить и оценить библиотечные функции для обработки строк. Привести примеры использования.
13. Охарактеризовать оператор присваивания в языке Python.
14. Охарактеризовать конструкцию ветвления: представление в блок схемах и языках программирования. Привести примеры полной и неполной формы.
15. Дать определение понятиям: компиляция, компоновка программы.
16. Дать характеристику локальным и глобальным переменным: область действия и область видимости.
17. Рассказать о данных числовых типов в Python и перечислить операции, допустимые для данных числовых типов.
18. Охарактеризовать типовые алгоритмы частичной обработки списков (по строкам или по столбцам).
19. Описать механизм использования библиотечных функций.
20. Описать многомерный список: объявление, инициализация, индексация элементов, формирование и вывод.
21. Охарактеризовать конструкцию выбора: представление в блок схемах и языках программирования. Привести пример
22. Описать конструкцию цикла: представление в блок схемах и языках программирования. Привести пример с использованием оператора for.
23. Охарактеризовать функции: понятие, объявление, прототипы функций. Вызов функции. Типы возвращаемых значений.

24. Описать конструкцию цикла while: представление в блок схемах и языках программирования.
25. Дать определение понятию «структура». Описать принцип работы со структурой.
26. Описать оператор условного перехода if-else (elif). Перечислить и описать его компоненты.
27. Строковые переменные и функции для работы с ним.
28. Перечислить основные алгоритмические конструкции. Описать оператор вложенного цикла. Привести пример.
29. Дать определение понятию «модуль». Перечислить и охарактеризовать методы разработки структурированных программ.
30. Дать определение понятиям «функция». Привести пример объявления функции, описать структуру функции.

Практическая часть

1. Написать программу для вычисления суммы элементов двух списков, введенных с клавиатуры.
2. Написать программу для вычисления произведения элементов списков, введенных с клавиатуры, с реализацией проверки на возможность выполнения операции.
3. Написать программу для сортировки списка.
4. Написать программу-калькулятор для вычисления суммы, разности, произведения и частного двух чисел с реализацией меню.
5. Написать программу для вычисления факториала введенного числа.
6. Написать программу для подсчета количества строк в файле.
7. Написать программу для подсчета количества символов в каждой строке файла.
8. Написать программу для замены табуляции в файле пробелами.
9. Написать программу для сортировки строк в файле.
10. Написать программу для копирования содержимого файла.
11. Написать программу-калькулятор для основных тригонометрических функций.
12. Программа просит пользователя ввести a и b, если $a > b > 5$, то программа выводит на экран слово "осень", если $a = b$, то выводит значение выражения $\frac{a+b}{a^2+b^2+1}$, во всех остальных случаях ввести c и вывести значение выражения $\sqrt{c} * \sin(c^2 + a^2b) - \frac{1}{7}$.
13. Напишите программу нахождения гипотенузы и площади прямоугольного треугольника по двум катетам.
14. Напишите программу вычисления среднего арифметического и среднего геометрического двух чисел.
15. Составить программу подсчета общего количества цифр и знаков "+", "-", ".", "*" в строке s, введенной с клавиатуры,
16. Составить программу печать элементов данного множества в алфавитном порядке
17. Составить программу формирования множества строчных латинских букв, входящих в строку, введенную с клавиатуры, и подсчета количества знаков препинания в ней.

**КРИТЕРИИ И ШКАЛА ОЦЕНИВАНИЯ ОТВЕТА ОБУЧАЮЩЕГОСЯ НА
ДИФФЕРЕНЦИРОВАННОМ ЗАЧЕТЕ**

<i>Оценка</i>	<i>Критерии оценки</i>
<i>Отлично</i>	Обучающийся владеет знаниями и умениями дисциплины в полном объеме рабочей программы, достаточно глубоко осмысливает дисциплину; самостоятельно, в логической последовательности и исчерпывающе отвечает на все вопросы зачетного билета, умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать, и систематизировать изученный материал, выделять в нем главное; устанавливать причинно-следственные связи; четко формирует ответы, решает задачи повышенной сложности.
<i>Хорошо</i>	Обучающийся владеет знаниями и умениями дисциплины почти в полном объеме программы (имеются пробелы знаний только в некоторых, особенно сложных разделах); самостоятельно и отчасти при наводящих вопросах дает полноценные ответы на вопросы билета; не всегда выделяет наиболее существенное, не допускает вместе с тем серьезных ошибок в ответах; умеет решать средней сложности задачи.
<i>Удовлетворительно</i>	Обучающийся владеет обязательным объемом знаний по дисциплине; проявляет затруднения в самостоятельных ответах, оперирует неточными формулировками; в процессе ответов допускаются ошибки по существу вопросов. Обучающийся способен решать лишь наиболее легкие задачи, владеет только обязательным минимумом знаний.
<i>Неудовлетворительно</i>	Обучающийся не освоил обязательного минимума знаний по дисциплине, не способен ответить на вопросы билета даже при дополнительных наводящих вопросах.

5.4. Вопросы и задания для проверки уровня сформированности компетенций

3 семестр

КОМПЕТЕНЦИЯ ОК 01

Перечень оценочных материалов (закрытого типа)

Номер вопроса	Вопрос
1.	Бинарный поиск информации основан на: а) сравнении некоторого ключа со средним элементом массива(списка) данных; б) поиске элемента в заданной структуре данных, пока не достигнет конца структуры
2.	Каким способом можно вычислить синус числа? а) использовать строку кода $x = \sin(a)$ б) подключить библиотеку <code>random</code> в) подключить библиотеку <code>math</code> г) подключить библиотеку <code>math</code> и использовать в коде функцию <code>sin</code>
3.	Как найти сумму 20 вводимых чисел с клавиатуры в Python? а) сложить все числа между собой б) использовать цикл на 20 повторений
4.	Какой из способов преобразует строковую переменную в целое число? а) <code>str(a)</code> б) <code>float(a)</code> в) <code>int(a)</code>
5.	Как пояснить отрывок кода не нарушая ход его работы в Python? а) в скобках рядом со строкой кода написать пояснение б) закомментировать строку с помощью <code>#</code> в) применить двухсторонний знак <code>!</code> в виде <code>: ! строка кода !</code>
6.	Какая строка позволяет вводить данные числового типа? а) <code>input(a)</code> б) <code>input(int)</code> в) <code>int(input())</code>
7.	Какой из строк кода позволяет увеличивать переменную на 2 единицы? а) <code>s = 2s</code>

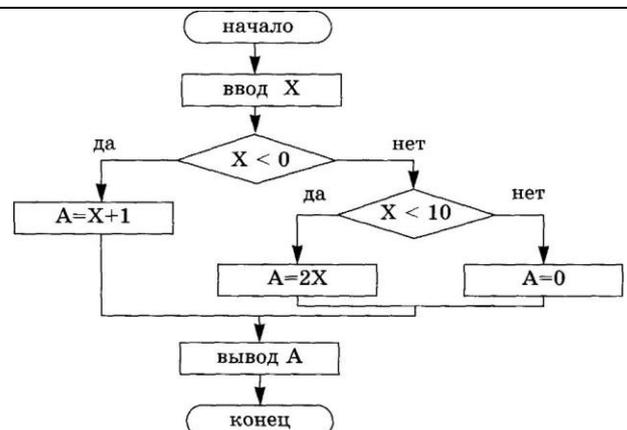
Номер вопроса	Вопрос
	б) $s = s+2$ с) $s = s*2$ д) $s = a +2$ е) $s +=2$
8.	Если работа программного кода должна позволять делать некий выбор в зависимости от условия как вид алгоритма необходимо использовать? а) линейный б) циклический в) разветвленный г) перемещения

Ключ ответов

№ вопроса	Верный ответ
1.	а
2.	г
3.	б
4.	с
5.	б
6.	с
7.	б, е
8.	в

Примерные задания оценочных материалов (открытого типа)

Номер задания	Содержание вопроса/задания
1.	Какую из циклических структур удобнее применять для решения следующей задачи: вывод значений функции $y=x^2 + \sin(x)$ на отрезке $[a,b]$ с шагом h
2.	Напишите результат работы данной блок-схемы при $X = 8$



Номер задания	Содержание вопроса/задания
3.	<p>Напишите результат работы данного программного кода:</p> <pre>w=3 p=5 j=8.5 if (j<p) and (j>w): j=j+0.5 j=j+10 else: j=11 print("j=", j)</pre>
4.	<p>Какой тип данных сохранится в переменной A при вводе информации строкой кода:</p> <pre>A = input()</pre>

Ключ ответов

№ вопроса	Верный ответ
1.	Ответ: <i>FOR</i>
2	16
3	j=11
4	строковый

КОМПЕТЕНЦИЯ ОК 02

Примерный перечень оценочных материалов (закрытого типа)

Номер вопроса	Вопрос
1.	<p>Определите ошибку в работе данного кода по выводимому сообщению интерпретатора:</p> <pre>1 a = 8 2 b = 10 3 c = a b</pre> <p>-----</p> <p>File "<ipython-input-8-3b3ffcedf995>", line 3 c = a b ^ SyntaxError: invalid syntax</p> <p>а) нет блочного отступа б) нет двоеточия в) нет знака умножения между</p>

Номер вопроса	Вопрос
	переменными а и б г) проблема с типами данных
2.	<p>Определите ошибку в работе данного кода по выводимому сообщению интерпретатора:</p> <pre> 1 for i in range(10): 2 print('Привет Мир!') ----- File "<ipython-input-6-628f419d2da8>", line 2 print('Привет Мир!') ^ IndentationError: expected an indented block </pre> <p>а) нет блочного отступа б) нет двоеточия в) нет кавычек в) не указан шаг цикла</p>
3.	<p>Определите ошибку в работе данного кода по выводимому сообщению интерпретатора:</p> <pre> 1 a = 2 2 b = 'PythonRu' 3 a + b ----- TypeError Traceback (most recent call last) <ipython-input-7-86a706a0ffdf> in <module>, line 3 1 a = 2 2 b = 'PythonRu' ----> 3 a + b </pre> <p>TypeError: unsupported operand type(s) for +: 'int' and 'str'</p> <p>а) нет блочного отступа б) нет двоеточия в) нет знака умножения между переменными а и б в) проблема с типами данных</p>

Ключ ответов

№ вопроса	Верный ответ
1.	в
2.	а
3.	в

Примерный перечень оценочных материалов
(открытого типа)

Номер задания	Содержание вопроса/задания
1.	<p>Какое значение будет находиться в ячейке z после выполнения группы операторов? Сделайте трассировку</p> <pre> x=100 z=10 x=x/z </pre>

Номер задания	Содержание вопроса/задания
	$x=x*z$ $z=z+30$
2.	Какое значение будет находиться в ячейке x после выполнения группы операторов? Сделайте трассировку $x=10$ $x=x+3$ $a=x+x$ $x=a$
3.	Какое значение будет находиться в ячейке a после выполнения группы операторов? Сделайте трассировку $a=26$ $a='Иванов'$ $b=a$ $b=54$ $b=0$ $a=b$
4.	Какое значение будет находиться в ячейке a после выполнения группы операторов? Сделайте трассировку $a=3.33$ $b=3.31$ $c=\min(a,b)$ $b=b+c$ $d=a$ $a=b$

Ключ ответов

№ вопроса	Верный ответ
1.	40
2.	26
3.	0
4.	6.62

КОМПЕТЕНЦИЯ ОК 09

Примерный перечень оценочных материалов (закрытого типа)

Номер вопроса	Вопрос
1.	Что такое IDE? а) компилятор б) оператор языка Python в) редактор программного кода
2.	В чем ошибка работы следующих строк кода? $a = \text{int}(\text{input}(\text{'Введите a: '}))$ $b = \text{int}(\text{input}(\text{'Введите b: '}))$

Номер вопроса	Вопрос
	<pre>for i in range(a,b): if i % 3 == 0: count += 1 kr_3 += i</pre> <p>IndentationError: expected an indented block after 'for' statement</p> <p>а) нет блочного отступа б) у цикла for отсутствует шаг итерации в) нет блочного отступа у оператора if</p>
3.	<p>Какой оператор цикла используют, если не известно количество повторов (итераций)?</p> <p>а) while б) for с) repeat d) break</p>
4.	<p>Какие операторы можно использовать, если необходимо несколько раз перемножить какое-нибудь число?</p> <p>а) for б) while в) input() г) write</p>
5.	<p>Что означает следующий программный код:</p> <pre>f == 0</pre> <p>а) переменной f присваивается 0 б) переменная f равна 0 в) переменная f равна переменной 0</p>

Ключ ответов

№ вопроса	Верный ответ
1.	в)
2.	а)
3.	а)
4.	а,б
5.	б

Примерный перечень оценочных материалов
(открытого типа)

Номер задания	Содержание вопроса/задания
1.	Дана строка из 5-ти символов: ЗАОГР. Что выведется на экран в результате работы данного кода: <pre>print(x[3]+x[4]+x[2]+x[0]+x[1])</pre>
2.	Используя знания о приоритетах арифметических вычислений, укажите полученный ответ: $11 * 2 ** 2 - 12 / 4 + 7$
3.	Какая команда используется для подключения библиотеки функций?
4.	Дан список животных: <pre>ANIMALS=['мышь','лошадь','курица','сова','змея','мартышка']</pre> Как вывести на экран слово «сова» из этого списка?
5.	Напишите программный код ввода числа А с клавиатуры?
6.	Что выведется на экран, если $a = 12$? <pre>a = int(input()) d = a % 10 print(d * 2)</pre>

Ключ ответов

№ вопроса	Верный ответ
1.	ГРОЗА
2.	48
3.	import
4.	print(ANIMALS[3])
5.	A=int(input())
6.	4

ВОПРОСЫ И ЗАДАНИЯ ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

4 семестр

КОМПЕТЕНЦИЯ ПК 2.5

Примерный перечень оценочных материалов (закрытого типа)

1. Укажите правильную структуру кода

а)

```
while True:
    ___number = int(input('Введите
число: '))
    ___count += 1
    ___if number == 7:
        _____print('Вы угадали!
Число попыток: ',count)
        ___break
    ___elif number < 7:
        _____print('Число меньше
чем нужно. Попробуйте снова')
```

б)

```
while True:
    __number = int(input('Введите
число: '))
    __count += 1
    __if number == 7:
        __print('Вы угадали! Число
попыток: ',count)
        __break
    __elif number < 7:
        __print('Число меньше чем
нужно. Попробуйте')
```

2. Укажите блочные операторы в Python

- a) for
- б) while
- с) if
- d) import
- e) true

3. Стандарт отступов в блочных элементах:

- a. Два пробела
- b. Один пробел
- с. Один Enter
- d. Один Tab
- e. Четыре пробела

4. Укажите предпочтительное написание комментариев:

- a. first_name = 'Ivan' # проверка имени пользователя на входе
- b. first_name = 'Ivan' # имя пользователя

5. Укажите предпочтительное написание имен переменных:

- a. first_name = 'Ivan'
- b. last_name = 'Ivanov'

- c. `fnm = 'Ivan'`
 - d. `lnm = 'Ivanov'`
6. Укажите правильное написание имен переменных:
 - a. `2023_year = 6.785`
 - b. `Year_2023 = 6.785`
 - c. `Year2023 = '6.785'`
 - d. `_Year2023 = 6.785`
 7. Какой метод отвечает за добавление элемента в конец списка?
 - a. `extend()`
 - b. `add()`
 - c. `include()`
 8. Какое ключевое слово используется в Python для создания класса?
 - a. `create`
 - b. `main`
 - c. `class`
 - d. `declare`
 9. Что представляет собой Python-список?
 - a. многомерный массив
 - b. позиционно упорядоченную коллекцию с произвольными типами элементов
 - c. упорядоченный массив
 - d. разновидность FIFO-очереди
 10. Какие типы данных допустимы в множестве?
 - a. любые
 - b. числовые
 - c. строковые и целочисленные
 - d. числовые и строковые
 11. Какой оператор отвечает за прерывание работы цикла в Python?
 - a. `continue`
 - b. `stop`
 - c. `break`
 - d. `try-except`
 12. Что нужно сделать для того, чтобы воспользоваться созданной реализацией?
 - a. создать экземпляр
 - b. вызвать имеющиеся методы напрямую
 - c. вызвать конструктор по умолчанию
 - d. создать класс для вызова
 13. Какое ключевое слово используется в Python для создания конструктора?
 - a. `__main__`
 - b. `__init__`
 - c. `__name__`
 - d. `__class__`

Ключ ответов

№ вопроса	Верный ответ
1.	a)
2.	a, б, с

3.	d,e
4.	a
5.	a,b
6.	b,c,d
7.	a)
8.	c
9.	b)
10.	a)
11.	c)
12.	a)
13.	b

КОМПЕТЕНЦИЯ ПК 09

Примерный перечень оценочных материалов (открытого типа)

1. Напишите программу, запрашивающую ввод списка чисел (или строки) и преобразующего его в множество
2. Напишите программу, запрашивающую ввод произвольных данных в список, и, для данных кратных 5 -формируется новый список, преобразующийся затем во множество.
3. Предоставлен список натуральных чисел. Требуется сформировать из них множество. Если какое-либо число повторяется, то преобразовать его в строку по образцу: например, если число 4 повторяется 3 раза, то в множестве будет следующая запись: само число 4, строка 44 (второе повторение, т.е. число дублируется в строке), строка 444 (третье повторение, т.е. строка множится на 3).
4. Дан список, состоящий из чисел. Получить множество из списка. Если число в списке повторяется, то вместо него в множество добавляется строка. Например в списке число 7. В множество добавляется число 7. Если оно встречается дважды, то в множество добавляется '*7'. Если трижды то '*7' и '**7'. И т.д.
5. Напишите функцию `sum_range(start, end)`, которая суммирует все целые числа от значения `start` до величины `end` включительно. Если пользователь задаст первое число большее чем второе, просто поменяйте их местами.
6. Чтобы проверить понимание параметров и область их видимости программист создал 3 функции (представлены ниже). Попробуйте предугадать, как поведет себя каждая из них при запуске (возникнут ли ошибки, что возвратится).

```

a)
def func1():
    param = 4

    def inner():
        param += 1

    return param

```

```

б)
def func2():
    _____param = 4

    def inner(var):
        var += 1

    inner(param)
    return param

```

```

в)
def func3():
    param = 4

    def inner(var):
        var += 1
        return var

    param = inner(param)
    return param

```

7. Создайте функцию `three_args()`, которая принимает 1, 2 или 3 строго ключевых параметра. В результате ее работы на печать в консоль выводятся значения переданных переменных, но только если они не равны `None`. Получим, например, следующее сообщение: Переданы аргументы: `var1 = 2, var3 = 10`.
8. У программиста проблема: его функция `time_now()` работает очень странно. Казалось бы, задача простая: показать текущее время с сообщением. Тем не менее, время не меняется. Код предоставлен ниже с примерами. Постарайтесь решить проблему незадачливого программиста.

```

from datetime import datetime
from time import sleep
def time_now(msg, *, dt=datetime.now()):
    print(msg, dt)

```

```

# Тесты
time_now('Сейчас такое время: ')
sleep(1)
time_now('Прошла секунда: ')
sleep(1)
time_now('Ничего не понимаю... ')

```

```

Результат выполнения
Сейчас такое время: 2021-03-14 15:48:55.117455
Прошла секунда: 2021-03-14 15:48:55.117455
Ничего не понимаю... 2021-03-14 15:48:55.117455

```

9. Создайте класс `Soda` (для определения типа газированной воды), принимающий 1 аргумент при инициализации (отвечающий за добавку к выбираемому лимонаду). В

этом классе реализуйте метод `show_my_drink()`, выводящий на печать Газировка и {ДОБАВКА} в случае наличия добавки, а иначе отобразится следующая фраза: Обычная газировка.

10. Программисту требуется проверить, возможно ли из представленных отрезков условной длины сформировать треугольник. Для этого он решил создать класс `TriangleChecker`, принимающий только положительные числа. С помощью метода `is_triangle()` возвращаются следующие значения (в зависимости от ситуации):
11. – Ура, можно построить треугольник!;
 12. – С отрицательными числами ничего не выйдет!;
 13. – Нужно вводить только числа!;
 14. – Жаль, но из этого треугольник не сделать.
15. Программист создал класс `KgToPounds` с параметром `kg`, куда передается определенное количество килограмм, а с помощью метода `to_pounds()` они переводятся в фунты. Чтобы закрыть доступ к переменной `kg` он реализовал методы `set_kg()` - для задания нового значения килограммов, `get_kg()` - для вывода текущего значения кг. Из-за этого возникло неудобство: нам нужно теперь использовать эти 2 метода для задания и вывода значений. Помогите ему переделать класс с использованием функции `property()` и свойств-декораторов. Код приведен ниже.

```
class KgToPounds:
```

```
    def __init__(self, kg):
        self.__kg = kg

    def to_pounds(self):
        return self.__kg * 2.205

    def set_kg(self, new_kg):
        if isinstance(new_kg, (int, float)):
            self.__kg = new_kg
        else:
            raise ValueError('Килограммы задаются только числами')

    def get_kg(self):
        return self.__kg
```

16. Программист решил создать класс `Nikola`, принимающий при инициализации 2 параметра: имя и возраст. Но на этом он не успокоился. Не важно, какое имя передаст пользователь при создании экземпляра, оно всегда будет содержать Николая. В частности - если пользователя на самом деле зовут Николаем, то с именем ничего не произойдет, а если его зовут, например, Максим, то оно преобразуется в Я не Максим, а Николай. Более того, никаких других атрибутов и методов у экземпляра не может

быть добавлено, даже если кто-то и вздумает так поступить (т.е. если некий пользователь решит прибавить к экземпляру свойство отчество или метод приветствие, то ничего у такого хитреца не получится).

17. Что будет выведено на экран в результате выполнения кода условия? а) 1, б) 2, в) 3, г) ошибка.

```
class Test:
    def __init__(self, x=1):
        self.x = x

t_1 = Test()
t_2 = Test(2)
print(t_1.x + t_2.x)
```

18. Что будет выведено на экран в результате выполнения кода условия? а) 25, б) 30, в) 55, г) ошибка

```
class Person:
    def set_age(self, age):
        self.age = age

ivan = Person(25)
ivan.set_age(30)
print(ivan.age)
```

19. Что будет выведено на экран в результате выполнения кода условия? а) {'name': 'ivan', 'age': 25}, б) {'age': 25}, в) {'name': 'ivan'}, г) ошибка.

```
class Person:
    def set_age(self, age):
        self.age = age

ivan = Person()
setattr(ivan, 'name', 'ivan')
ivan.set_age(25)
print(ivan.__dict__)
del ivan.age
```

