

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет информатики и информационных технологий

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Технологии и методы программирования

Кафедра информационных систем и технологий программирования

Образовательная программа
09.03.03 Прикладная информатика

Направленность (профиль) программы
Информационные системы и программирование
Прикладная информатика в экономике и управлении
Прикладная информатика в юриспруденции

Уровень высшего образования
бакалавриат

Форма обучения
Очная

Статус дисциплины:
входит в обязательную часть ОПОП

Махачкала, 2022

Рабочая программа дисциплины “Технологии и методы программирования” составлена в 2022 году в соответствии с требованиями ФГОС ВО - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика от «19» сентября 2017г. № 922.

Разработчик(и): кафедра информационных систем и технологий программирования, доц. Баммаева Г.А.

Рабочая программа дисциплины одобрена:
на заседании кафедры ИСиТП от «01» марта 2022г., протокол № 8

Зав. кафедрой  Исмиханов З.Н.
(подпись)

на заседании Методической комиссии факультета ИиИТ
от «17» марта 2022г., протокол № 7

Председатель  Бакмаев А.Ш.
(подпись)

Рабочая программа дисциплины согласована с учебно-методическим управлением
«31» марта 2022г.

Начальник УМУ  Гасангаджиева А.Г.

Аннотация рабочей программы дисциплины

Дисциплина «Технологии и методы программирования» входит в обязательную часть, образовательной программы бакалавриата, по направлению 09.03.03 Прикладная информатика.

Дисциплина реализуется на факультете информатики и информационных технологий кафедрой информационных систем и технологий программирования.

Содержание дисциплины охватывает круг вопросов, связанных с формированием алгоритмического мышления у студентов, объектно-ориентированным программированием, созданием консольных и графических приложений.

Дисциплина нацелена на формирование следующих компетенций выпускника: общепрофессиональных – ОПК-7.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: лекции, практические занятия, лабораторные занятия, самостоятельная работа.

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости в форме контрольных работ, коллоквиума и промежуточный контроль в форме экзамена во втором и третьем семестрах.

Объем дисциплины 8 зачетных единиц, в том числе 288 академических часа по видам учебных занятий:

Очная форма обучения

Семестр	Учебные занятия							СРС, в том числе экзамен	Форма промежуточной аттестации (зачет, дифференциро ванный зачет, экзамен)
	в том числе:								
	всего	Контактная работа обучающихся с преподавателем							
		всего	из них						
	Лекции		Лаборат орные занятия	Практич еские занятия	КСР	консуль тации			
2	144	54	18	18	18			90	зачет
3	144	54	18	18	18			90	экзамен
Итого	288	108	36	36	36			180	

1. Цели освоения дисциплины

Целями освоения дисциплины (модуля) «Технологии и методы программирования» являются овладение знаниями в области технологии программирования; подготовка к осознанному использованию, как языков программирования, так и методов программирования. Формирование у студентов научного, творческого подхода к освоению технологий, методов и средств производства программного обеспечения. Получение необходимых знаний, умений и навыков в области применения современной вычислительной техники для решения практических задач обработки данных, математического моделирования, информатики, получение высшего профессионального (на уровне бакалавра) образования, позволяющего выпускнику успешно работать в избранной сфере деятельности с применением современных компьютерных технологий.

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина «Технологии и методы программирования» входит в обязательную часть, образовательной программы бакалавриата, по направлению 09.03.03 Прикладная информатика.

Для успешного освоения данного курса студент должен иметь элементарные знания по школьному курсу дисциплины «Информатика».

Список дисциплин, для изучения которых необходимы знания данного курса

1. Объектно-ориентированное программирование
2. Разработка программных приложений
3. Операционные системы
4. Программная инженерия

3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения и процедура освоения).

Код и наименование компетенции из ОПОП	Код и наименование индикатора достижения компетенций (в соответствии с ОПОП)	Планируемые результаты обучения	Процедура освоения
ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ИД 7.1. ОПК-7.1. Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий. ИД 7.2. ОПК-7.2. Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. ИД 7.3. ОПК-7.3. Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач.	<i>Знает:</i> методы разработки алгоритмов и программ, основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов. <i>Умеет:</i> разрабатывать алгоритмы и программы, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов <i>Владеет:</i> способами разработки алгоритмов и программ, основами информатики и программирования, применять их к проектированию, конструированию и тестированию программных продуктов.	Опрос, тестирование, контрольная работа

4. Объем, структура и содержание дисциплины.

Объем дисциплины составляет 8 зачетных единиц, 288 академических часов.

Структура дисциплины.

Структура дисциплины в очной форме

№	Разделы и темы дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)	
				Лекции	Практические	Лабораторные	Самостоятельная работа		
Семестр 2.									
Модуль 1. Основы языка программирования Python									
1.	Языки программирования и их назначение. Краткая история языков программирования	2		2	2	2	10	опрос, тестирование, контрольная работа, выполнение индивидуального задания	
2.	Знакомство с Python.	2		2	2	2	10		
3.	Ввод и вывод данных.	2		2	2	2	10		
Итого по модулю 1		2		6	6	6	30		
Модуль 2. Реализация линейных и разветвляющихся алгоритмов									
1.	Логические выражения и операторы	2		2	2	2	10	опрос, тестирование, контрольная работа, выполнение индивидуального задания	
2.	Линейные алгоритмы и программы.	2		2	2	2	10		
3.	Ветвление. Условный оператор. Множественное ветвление	2		2	2	2	10		
Итого по модулю 2		2		6	6	6	30		
Модуль 3. Реализация циклических алгоритмов.									
1.	Циклы в программировании. Цикл while	2		2	2	2	10	опрос, тестирование, контрольная работа, выполнение индивидуального задания	
2.	Цикл for. Использование счётчика. Функция range. Операторы break и continue.	2		2	2	2	10		
3.	Функции в программировании	2		2	2	2	10		
Итого по модулю 3		2		6	6	6	30		
Итого за семестр.					18	18	18	90	
Семестр 3. Программирование на C++									
Модуль 1. Основы программирования.									
1.	Алгоритмизация задачи. Понятие алгоритма. Свойства алгоритма.	3		3	3	3	9	опрос, тестирование, контрольная работа,	

2.	Базовые конструкции языка C++, простые типы данных, структура программы. Операторы языка.	3		3	3	3	9	выполнение индивидуального задания
Итого по модулю 1:				6	6	18	20	
Модуль 2. Структурированные типы данных.								
1.	Структурированные типы данных. Работа с массивами. Алгоритмы сортировки.	3		2	2	2	6	опрос, тестирование, контрольная работа, выполнение индивидуального задания
2.	Строковый тип.	3		2	2	2	6	
3.	Множества. Записи.	3		2	2	2	6	
Итого по модулю 2:				6	6	6	18	
Модуль 3. Работа с памятью.								
1.	Подпрограммы	3		2	2	2	6	опрос, тестирование, контрольная работа, выполнение индивидуального задания
2.	Работа с файлами	3		2	2	2	6	
3.	Процедуры и функции модуля Graph	3		2	2	2	6	
Итого по модулю 3:				6	6	6	18	
Модуль 4. Подготовка к экзамену.								
	Подготовка и сдача экзамена.						36	Экзамен
ИТОГО 3 сем:		144		18	18	18	90	
ИТОГО:		288		36	36	36	180	

Содержание дисциплины, структурированное по темам (разделам).

№ п/п	Наименование темы	Трудоемкость	Содержание	Формируемые компетенции	Результаты освоения (знать, уметь, владеть)	Технологии обучения
2 семестр						
Лекционный курс						
1.	Языки программирования и их назначение. Краткая история языков программирования	2	История развития языков программирования. Сравнительный обзор высокоуровневых языков программирования и общие понятия высокоуровневых языков программирования. Движущие силы развития языков программирования. Классификация языков программирования.	ОПК-7.1	Знает методы разработки алгоритмов и программ, основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов.	Интерактивная лекция, обсуждение
2.	Знакомство с Python.	2	Краткая историческая справка и основные особенности языка. Дзен Питона. Режимы работы с интерпретатором. Создание скриптов.	ОПК-7.1	Знает методы разработки алгоритмов и программ, основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов.	Интерактивная лекция, обсуждение
3.	Ввод и вывод данных	2	Данные и их типы. Операции в программировании. Изменение типов данных.	ОПК-7.1	Знает методы разработки алгоритмов и программ, основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов.	Интерактивная лекция, обсуждение
4.	Логические выражения и операторы	2	Логические выражения и логический тип данных. Логические операторы. Сложные логические выражения	ОПК-7.1	Знает методы разработки алгоритмов и программ, основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов.	Интерактивная лекция, обсуждение
5.	Линейные алгоритмы и программы	2	Понятие линейных алгоритмов. Построение линейных алгоритмов	ОПК-7.1	Знает методы разработки алгоритмов и программ, основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов.	Интерактивная лекция, обсуждение
6.	Ветвление. Условный оператор. Множественное	2	Понятие ветвления в языках программирования. Организация ветвлений в Python. Организация	ОПК-7.1	Знает методы разработки алгоритмов и программ, основы информатики и программирования, проектирования,	Интерактивная лекция, обсуждение

№ п/п	Наименование темы	Трудоемкость	Содержание	Формируемые компетенции	Результаты освоения (знать, уметь, владеть)	Технологии обучения
	ветвление		множественных ветвлений. Синтаксис if-elif-else. Вложенные условные конструкции. Каскадные условные конструкции.		конструирования и тестирования программных продуктов.	
7.	Циклы в программировании. Цикл while	2	Знакомство с организацией циклов в Python. Знакомство и применение цикла while.	ОПК-7.1	Знает методы разработки алгоритмов и программ, основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов.	Интерактивная лекция, обсуждение
8.	Цикл for. Использование счётчика. Функция range. Операторы break и continue	2	Знакомство и применение цикла for, функции range, операторов break и continue	ОПК-7.1	Знает методы разработки алгоритмов и программ, основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов.	Интерактивная лекция, обсуждение
9.	Функции в программировании	3	Понятие функции. Определение функции. Оператор def. Вызов функции. Структура программ с функциями.	ОПК-7.1	Знает методы разработки алгоритмов и программ, основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов.	Интерактивная лекция, обсуждение
Семестр 3. Программирование на C++						
1	Алгоритмизация задачи. Понятие алгоритма. Свойства алгоритма.	3	Понятие алгоритма и его свойства Способы описания алгоритмов	ОПК-7.1	Знает методы разработки алгоритмов и программ, основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов.	Интерактивная лекция, обсуждение
2	Базовые конструкции и языка C++, простые типы данных, структура программы. Операторы языка.	3	Структура и этапы создания программы на языке C++ Стандарты языка C++ Представление данных в языке C++	ОПК-7.1	Знает методы разработки алгоритмов и программ, основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов.	Интерактивная лекция, обсуждение
3	Структурированные типы	2	Массив, Размерность массива Описание массива	ОПК-7.1	Знает методы разработки алгоритмов и программ, основы информатики и	Интерактивная лекция,

№ п/п	Наименование темы	Трудоемкость	Содержание	Формируемые компетенции	Результаты освоения (знать, уметь, владеть)	Технологии обучения
	данных. Работа с массивами. Алгоритмы сортировки.		Типовые задачи обработки одномерных массивов.		программирования, проектирования, конструирования и тестирования программных продуктов.	обсуждени е
4	Строковый тип.	2	Строковые и символьные литералы (C++)	ОПК-7.1	Знает методы разработки алгоритмов и программ, основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов.	Интеракти вная лекция, обсуждени е
5	Множества Записи.	2	Составные типы данных	ОПК-7.1	Знает методы разработки алгоритмов и программ, основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов.	Интеракти вная лекция, обсуждени е
6	Подпрограммы	2	Общие сведения о функциях. Локальные и глобальные переменные	ОПК-7.1	Знает методы разработки алгоритмов и программ, основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов.	Интеракти вная лекция, обсуждени е
7	Работа с файлами	2	Функции работы с файлами и операции ввода-вывода	ОПК-7.1	Знает методы разработки алгоритмов и программ, основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов.	Интеракти вная лекция, обсуждени е
8	Процедуры и функции модуля Graph	2	Стандартный модуль Graph. Инициализация графического режима	ОПК-7.1	Знает методы разработки алгоритмов и программ, основы информатики и программирования, проектирования, конструирования и тестирования программных продуктов.	Интеракти вная лекция, обсуждени е

Лабораторные занятия

1	Линейные алгоритмы и программы	2	Программирование алгоритмов линейной структуры	ОПК-7.2, ОПК-7.3	Умеет разрабатывать алгоритмы и программы, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов. Владеет способами разработки алгоритмов и программ, основами информатики и программирования, применять их к проектированию, конструированию и тестированию программных продуктов	Защита лабораторных заданий, выполнение индивидуального задания
2	Ветвление Условный оператор. Множественное ветвление	2	Организация ветвлений в Python. Организация множественных ветвлений. Синтаксис if-elif-else. Вложенные условные конструкции. Каскадные условные конструкции.	ОПК-7.2, ОПК-7.3	Умеет разрабатывать алгоритмы и программы, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов. Владеет способами разработки алгоритмов и программ, основами информатики и программирования, применять их к проектированию, конструированию и тестированию программных продуктов	Защита лабораторных заданий, выполнение индивидуального задания
3	Ошибки и исключения. Обработка исключений	2	Обработка исключений. Оператор try-except.	ОПК-7.2, ОПК-7.3	Умеет разрабатывать алгоритмы и программы, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов. Владеет способами разработки алгоритмов и программ, основами информатики и программирования, применять их к проектированию, конструированию и тестированию программных продуктов	Защита лабораторных заданий, выполнение индивидуального задания
4	Циклы в программах	2	Применение цикла while.	ОПК-7.2,	Умеет разрабатывать алгоритмы и программы,	Защита лабораторных заданий,

	ировании. Цикл while			ОПК-7.3	применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов. Владеет способами разработки алгоритмов и программ, основами информатики и программирования, применять их к проектированию, конструированию и тестированию программных продуктов	ных заданий, выполнен ие индивидуального задания
5	Цикл for. Использование счётчика. Функция range. Операторы break и continue	2	Работа с циклом for. Функция range(). Цикл for и range(). Операторы break и continue	ОПК-7.2, ОПК-7.3	Умеет разрабатывать алгоритмы и программы, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов. Владеет способами разработки алгоритмов и программ, основами информатики и программирования, применять их к проектированию, конструированию и тестированию программных продуктов	Защита лабораторных заданий, выполнен ие индивидуального задания
6	Функции в программировании	2	Оператор def. Вызов функции. Структура программ с функциями.	ОПК-7.2, ОПК-7.3	Умеет разрабатывать алгоритмы и программы, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов. Владеет способами разработки алгоритмов и программ, основами информатики и программирования, применять их к проектированию, конструированию и тестированию программных продуктов	Защита лабораторных заданий, выполнен ие индивидуального задания
7	Локальные и глобальные переменные	2	Работа с локальными и глобальными переменными в Python.	ОПК-7.2, ОПК-7.3	Умеет разрабатывать алгоритмы и программы, применять основы информатики и программирования к	Устный опрос, тестирование, защита

	ые				проектированию, конструированию и тестированию программных продуктов. Владеет способами разработки алгоритмов и программ, основами информатики и программирования, применять их к проектированию, конструированию и тестированию программных продуктов	лабораторной работы, выполнение индивидуального задания
8	Возврат значений из функций. Оператор return	2	Знакомство с функцией return. Возврат нескольких значений.	ОПК-7.2, ОПК-7.3	Умеет разрабатывать алгоритмы и программы, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов. Владеет способами разработки алгоритмов и программ, основами информатики и программирования, применять их к проектированию, конструированию и тестированию программных продуктов	Устный опрос, тестирование, защита лабораторной работы, выполнение индивидуального задания

4.3.2. Содержание лабораторно-практических занятий по дисциплине

Лабораторная работа: Работа со списками. Операции над списками в Python

Цель работы: Изучение одномерных массивов в Python.

Массивы (списки) в Python - это определенное количество элементов одного типа, которые имеют общее имя, и каждый элемент имеет свой индекс - порядковый номер.

Часто для работы с массивами используются списки.

Список (list) - это структура данных для хранения объектов различных типов.

Списки являются упорядоченными последовательностями, которые состоят из различных типов данных, заключающихся в квадратные скобки [] и отделяющиеся друг от друга с помощью запятой.

Создание списков на Python.

Создать список можно несколькими способами

1. Получение списка через присваивание конкретных значений.

Так выглядит в коде Python пустой список:

```
s = [] # Пустой список
```

Примеры создания списков со значениями:

```
l=[5, 75, -4, 7, -51] # список целых чисел
l=[1.13, 5.34, 12.63, 4.6, 34.0, 12.8] # список из вещественных чисел
l=["Оля", "Владимир", "Михаил", "Дарья"] # список из строк
l=["Москва", "Иванов", 12, 124] # смешанный список
l=[[0, 0, 0], [1, 0, 1], [1, 1, 0]] # список, состоящий из списков
l=['s', 'p', ['isok'], 2] # список из значений и списка
```

Списки можно складывать (конкатенировать) с помощью знака «+»:

```
l=[1, 3]+[4, 23]+[5]
print('l=[1, 3]+[4, 23]+[5] =', l)
```

Результат:

```
>>>
l=[1, 3]+[4, 23]+[5] = [1, 3, 4, 23, 5]
>>> |
```

2. Создание списка при помощи функции Split().

Используя функцию split в Python можно получить из строки список.

```
stroka="Привет, страна"
```

```
lst=stroka.split(",")
```

```
stroka="Здравствуй, Дедушка Мороз" #stroka - строка
lst=stroka.split(",") #lst - список
print('stroka = ', stroka)
print('lst=stroka.split(","):', lst)
```

Результат

```
==== RESTART: C:/Users/maxim/Desktop/ex_list_
stroka = Здравствуй, Дедушка Мороз
lst=stroka.split(","): ['Здравствуй', ' Дедушка Мороз']
```

3. Генераторы списков.

В Python создать список можно также при помощи генераторов.

Первый способ.

Сложение одинаковых списков заменяется умножением:

список из 10 элементов, заполненный единицами

```
l = [1]*10
```

Второй способ.

Пример 1.

```
l = [i for i in range(10)]
```

Пример 2.

```
c = [c * 3 for c in 'list']
```

```
print(c) # ['lll', 'iii', 'sss', 'ttt']
```

```
Создание списка из строки.
l = list (строка):
['c', 'т', 'р', 'о', 'к', 'а']
```

```
Создание списка при помощи функции Split().
stroka=" Hello, friend "
lst=stroka.split(","):
['Hello', ' friend']
```

```
Генераторы списков.
Первый способ.
l = [1]*10:
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

```
Второй способ. Пример 1.
l = [i for i in range(10)]:
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
Второй способ. Пример 2.
c=[c*3 for c in "list"]:
['lll', 'iii', 'sss', 'ttt']
```

Примеры использования генераторов списка.

Пример 1.

Заполнить список квадратами чисел от 0 до 9, используя генератор списка.

Решение:

```
l = [i*i for i in range(10)]
```

Пример 2.

Заполнить список числами, где каждое последующее число больше на 2.

```
l = [(i+1)+i for i in range(10)]
```

```
print(l)
```

Заполнить список квадратами чисел от 0 до 9, используя генератор списка.

```
l = [i*i for i in range(10)]:
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

Заполнить список числами, где каждое последующее число больше на 2.

```
l = [(i+1)+i for i in range(10)]:
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
```

Модуль random предоставляет функции для генерации случайных чисел, букв, случайного выбора элементов последовательности.

random.randint(A, B) - случайное целое число N, $A \leq N \leq B$.

random.random() - случайное число от 0 до 1.

Случайные числа в списке:

10 чисел, сгенерированных случайным образом в диапазоне (10,80)

```
from random import randint
```

```
l = [randint(10,80) for x in range(10)]
```

10 чисел, сгенерированных случайным образом в диапазоне (0,1)

```
l = [random() for i in range(10)]
```

```
from random import *
l = [randint(10,80) for i in range(10)]
print('10 чисел, сгенерированных случайным образом в диапазоне (10,80).')
print('l = [randint(10,80) for x in range(10):']
print(l)
print()

l = [random() for i in range(10)]
print('10 чисел сгенерированных в диапазоне от 0 до 1.')
print('l = [random() for i in range(10):']
for i in range(len(l)):
    print('{:.2f}'.format(l[i]), end = " ")
```

```
10 чисел, сгенерированных случайным образом в диапазоне (10, 80).
l = [randint(10,80) for x in range(10)]:
[70, 33, 79, 61, 34, 27, 11, 55, 52, 31]
```

```
10 чисел сгенерированных в диапазоне от 0 до 1.
l = [random() for i in range(10)]:
0.66 0.97 0.87 0.57 0.54 0.83 0.57 0.65 0.04 0.07
```

4. Ввод списка (массива) в языке Python.

Для ввода элементов списка используется цикл for и команда range ():

```
for i in range(N):
    x[i] = int(input())
```

Более простой вариант ввода списка:

```
x = [int(input()) for i in range(N)]
```

```
print('Ввод списка. Пример 1:')
x=[]
for i in range(4):
    x.append(int(input()))
print(x)

x=[]
print('Ввод списка. Пример 2:')
x = [int(input()) for i in range(4)]
print(x)
```

Ввод списка. Пример 1:

```
45
4
85
2
```

```
[45, 4, 85, 2]
```

Ввод списка. Пример 2:

```
4
5
7
8
```

```
[4, 5, 7, 8]
```

Функция int здесь используется для того, чтобы строка, введенная пользователем, преобразовывалась в целые числа.

5. Вывод списка (массива) в языке Python.

Вывод целого списка (массива):

```
print(L)
```

Поэлементный вывод списка (массива):

```
for i in range(N):
    print(L[i], end = " ")
```

```
Вывод целого списка (массива)
[1, 56, 6, 3, 6, 7, 3, 37, 7, 37, 37]
```

```
Поэлементный вывод списка (массива)
1 56 6 3 6 7 3 37 7 37 37
```

Методы списков

Метод	Что делает
<code>list.append(x)</code>	Добавляет элемент в конец списка
<code>list.extend(L)</code>	Расширяет список list, добавляя в конец все элементы списка L
<code>list.insert(i, x)</code>	Вставляет перед i-ым элементом значение x
<code>list.remove(x)</code>	Удаляет первый элемент в списке, имеющий значение x. ValueError, если такого элемента не существует

list.pop(i)	Удаляет i-ый элемент и возвращает его. Если индекс не указан, удаляется последний элемент
list.index(x, [start [, end]])	Возвращает положение первого элемента со значением x (при этом поиск ведется от start до end)
list.count(x)	Возвращает количество элементов со значением x
list.reverse()	Разворачивает список
list.copy()	Поверхностная копия списка
list.clear()	Очищает список

Ниже приведена программа, демонстрирующая методы работы списков.

```

a=[0, 2, 2, 2, 4] #список a
b=[5, 6, 7, 2, 9] #список b
print ('Исходный список a:', a)
print ('Исходный список b:', b)
x=99
y=5

a.append(x)
print ('a.append(x):', a)

a.extend(b)
print ('a.extend(b):', a)

a.insert(3, x)
print ('a.insert(3, x):', a)

a.remove(x)
print ('a.remove(x):', a)

print ('a.pop(5):', a.pop(5))
print (a)

print ('a.index(y, 0, len(a)):', a.index(y, 0, len(a)))

print ('a.count(2):', a.count(2))

a.reverse()
print ('a.reverse():', a)

z=a.copy()
print ('z=a.copy():', z)

z.clear()
print ('z.clear():')
print ('z =', z)

```

Пример программы на Python

```

Исходный список a: [0, 2, 2, 2, 4]
Исходный список b: [5, 6, 7, 2, 9]
a.append(x): [0, 2, 2, 2, 4, 99]
a.extend(b): [0, 2, 2, 2, 4, 99, 5, 6, 7, 2, 9]
a.insert(3, x): [0, 2, 2, 99, 2, 4, 99, 5, 6, 7, 2, 9]
a.remove(x): [0, 2, 2, 2, 4, 99, 5, 6, 7, 2, 9]
a.pop(5): 99
[0, 2, 2, 2, 4, 5, 6, 7, 2, 9]
a.index(y, 0, len(a)): 5
a.count(2): 4
a.reverse(): [9, 2, 7, 6, 5, 4, 2, 2, 2, 0]
z=a.copy(): [9, 2, 7, 6, 5, 4, 2, 2, 2, 0]
z.clear():
z = []

```

Результат выполнения программы

Пример выполнения задания

Вариант 0

1. Из массива X длиной n, среди элементов которого есть положительные, отрицательные и равные нулю, сформировать новый массив Y, взяв в него только те элементы из X, которые больше по модулю заданного числа M. Вывести на экран число M, данный и полученные массивы.

Решение:


```

n=int(input('Введите длину массива\n'))
m=int(input('Введите число M\n'))
x=[]
y=[]
for i in range(n):
    print('Введите ', i, 'элемент:')
    x.append(int(input()))
for i in range(n):
    if abs(x[i])>m:
        y.append(x[i])
print('Введённое число M:', m)
print('Массив X:', x)
print('Массив Y:', y)

```

```

Введите длину массива
5
Введите число M
20
Введите 0 элемент:
21
Введите 1 элемент:
22
Введите 2 элемент:
5
Введите 3 элемент:
6
Введите 4 элемент:
8
Введённое число M: 20
Массив X: [21, 22, 5, 6, 8]
Массив Y: [21, 22]

```

Задания для самостоятельного выполнения

Вариант 1

1. Дан одномерный массив, состоящий из N целочисленных элементов. Ввести массив с клавиатуры. Найти максимальный элемент. Вывести массив на экран в обратном порядке.
2. В массиве действительных чисел все нулевые элементы заменить на среднее арифметическое всех элементов массива.

Вариант 2

1. Дан одномерный массив, состоящий из N целочисленных элементов. Ввести массив с клавиатуры. Найти минимальный элемент. Вывести индекс минимального элемента на экран.
2. Дан массив целых чисел. Переписать все положительные элементы во второй массив, а остальные - в третий.

Вариант 3

1. В одномерном числовом массиве D длиной n вычислить сумму элементов с нечетными индексами. Вывести на экран массив D, полученную сумму.
2. Дан одномерный массив из 8 элементов. Заменить все элементы массива меньшие 15 их удвоенными значениями. Вывести на экран монитора преобразованный массив.

Вариант 4

1. Дан массив целых чисел. Найти максимальный элемент массива и его порядковый номер.
2. Дан одномерный массив целого типа. Получить другой массив, состоящий только из нечетных чисел исходного массива или сообщить, что таких чисел нет. Полученный массив вывести в порядке убывания элементов.

Вариант 5

1. Дан одномерный массив из 10 целых чисел. Вывести пары отрицательных чисел, стоящих рядом.
2. Дан целочисленный массив размера 10. Создать новый массив, удалив все одинаковые элементы, оставив их 1 раз.

Вариант 6

1. Дан одномерный массив из 10 целых чисел. Найти максимальный элемент и сравнить с ним остальные элементы. Вывести количество меньших максимального и больших максимального элемента.
2. Одномерный массив из 10-и целых чисел заполнить с клавиатуры, определить сумму тех чисел, которые >5.

Вариант 7

1. Дан массив целых чисел. Найти сумму элементов с четными номерами и произведение элементов с нечетными номерами. Вывести сумму и произведение.
2. Переставить в одномерном массиве минимальный элемент и максимальный.

Вариант 8

1. Найдите сумму и произведение элементов списка. Результаты вывести на экран.
2. В массиве действительных чисел все нулевые элементы заменить на среднее арифметическое всех элементов массива.

Вариант 9

1. Дан одномерный массив, состоящий из N вещественных элементов. Ввести массив с клавиатуры. Найти и вывести минимальный по модулю элемент. Вывести массив на экран в обратном порядке.
2. Даны массивы A и B одинакового размера 10. Вывести исходные массивы. Поменять местами их содержимое и вывести в начале элементы преобразованного массива A, а затем - элементы преобразованного массива B.

Вариант 10

1. Определите, есть ли в списке повторяющиеся элементы, если да, то вывести на экран это значение, иначе сообщение об их отсутствии.
2. Дан одномерный массив из 15 элементов. Элементам массива меньше 10 присвоить нулевые значения, а элементам больше 20 присвоить 1. Вывести на экран монитора первоначальный и преобразованный массивы в строку.

Вариант 11

1. Найти наибольший элемент списка, который делится на 2 без остатка и вывести его на экран.
2. Дан одномерный массив целого типа. Получить другой массив, состоящий только из четных чисел исходного массива, меньше 10, или сообщить, что таких чисел нет. Полученный массив вывести в порядке возрастания элементов.

Вариант 12

1. Найти наименьший нечетный элемент списка и вывести его на экран.
2. Даны массивы A и B одинакового размера 10. Поменять местами их содержимое и вывести вначале элементы преобразованного массива A, а затем - элементы преобразованного массива B.

Вариант 13

1. Дан одномерный массив целых чисел. Проверить, есть ли в нем одинаковые элементы. Вывести эти элементы и их индексы.
2. Дан одномерный массив из 8 элементов. Заменить все элементы массива меньше 15 их удвоенными значениями. Вывести на экран монитора преобразованный массив.

Вариант 14

1. Найти максимальный элемент численного массива и поменять его местами с минимальным.
2. Программа заполняет одномерный массив из 10 целых чисел числами, считанными с клавиатуры. Определить среднее арифметическое всех чисел массива. Заменить элементы массива большие среднего арифметического на 1.

Вариант 15

1. Определите, есть ли в списке повторяющиеся элементы, если да, то вывести на экран эти значения.
2. Дан одномерный массив целого типа. Получить другой массив, состоящий только из нечетных чисел исходного массива или сообщить, что таких чисел нет. Полученный массив вывести в порядке убывания элементов.

5. Образовательные технологии

Образовательные технологии, используемые при изучении курса, предусматривают применение инновационных методов обучения. Использование персональных компьютеров при выполнении лабораторных работ и сдаче итогового экзамена. Чтение лекций с использованием компьютера и проектора, проведение лабораторных работ в компьютерном классе с доступом к интернету.

При реализации учебной дисциплины используются электронные практикумы, электронные учебники, презентации средства диагностики и контроля разработанные специалистами кафедры т.д.

Удельный вес занятий, проводимых в интерактивной форме, составляет 20% аудиторных занятий.

6. Учебно-методическое обеспечение самостоятельной работы студентов

Форма контроля и критерий оценок

В соответствии с учебным планом предусмотрен зачет во втором семестре и экзамен в третьем семестре.

Формы контроля: текущий контроль, промежуточный контроль по модулю, итоговый контроль по дисциплине предполагают следующее распределение баллов.

Примерное распределение времени самостоятельной работы студентов

Вид самостоятельной работы	Примерная трудоёмкость, в.ч.	Формируемые компетенции
	очная	
Текущая СРС		
работа с лекционным материалом, с учебной литературой	20	ОПК-7
опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	22	ОПК-7
самостоятельное изучение разделов дисциплины	20	ОПК-7
выполнение домашних заданий, домашних контрольных работ	20	ОПК-7
подготовка к лабораторным работам, к практическим и семинарским занятиям	22	ОПК-7
подготовка к контрольным работам, коллоквиумам, экзамену	36	ОПК-7
Творческая проблемно-ориентированная СРС		
выполнение расчётно-графических работ	10	ОПК-7
поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	10	ОПК-7
исследовательская работа, участие в конференциях, семинарах, олимпиадах	10	ОПК-7
анализ данных по заданной теме, выполнение расчётов, составление схем и моделей на основе собранных данных	10	ОПК-7
Итого СРС:	180	

7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

7.1. Типовые контрольные задания

Примерный перечень вопросов к промежуточному контролю или экзамену по всему изучаемому курсу:

1. Алгоритмы перевода чисел из одной системы счисления в другие. Перевод чисел в системах счисления с кратными основаниями
2. Арифметические действия в двоичной системе счисления
3. Этапы решения задач. Понятие алгоритма
4. Свойства и формы записи алгоритмов
5. Линейные алгоритмы. Ветвления в алгоритмах
6. Циклические алгоритмы
7. История развития языков программирования
8. Движущие силы развития языков программирования
9. Классификация языков программирования
10. Основные понятия алгоритмических языков программирования
11. Встроенный тип str. Методы объекта str.
12. print() и форматирование вывода.
13. Работа с файловой системой средствами Python.
14. Работа с файлами. Методы open(), close(), read(), write().
15. Модуль re. Синтаксис регулярных выражений, метасимволы.
16. Встроенные типы последовательностей list, tuple, range и их методы.
17. Встроенный объект dict и его методы.
18. Встроенные типы чисел -int, float, complex. Машинное представление чисел с плавающей точкой и целых. Преобразование типов при сравнении чисел.
19. Двоичное представление чисел. Неассоциативность операций в арифметике с плавающей запятой. Целые числа с произвольной точностью.
20. Множества. Встроенные типы set и frozenset.
21. Инструкции и синтаксис. Составные конструкции и обработка исключений
22. Инструкции if/else/elif, логические операторы и выражения сравнения
23. Циклы while и for в Python
24. Функции в Python. Основные понятия
25. Области видимости и пространство имен в Python.
26. Передача аргументов в функцию. Специальные режимы сопоставления аргументов.
27. Парадигма объектно-ориентированного программирования. Поддержка в Python функционального программирования.
28. Объекты. Динамическая типизация. Инкапсуляция.
29. Генерация объекта class. Новое пространство имен. Объект экземпляр класса.

30. Атрибуты класса. Атрибуты данных. Атрибуты-методы. Параметр self. Добавление атрибутов к классу во время исполнения программы.
31. Специальные методы и атрибуты классов. Методы `__init__()` и `__del__()` в Python. Декораторы функций и декораторы классов. Инструменты интроспекции в Python. Метаклассы.
32. Абстрактные методы в Python. Классические классы и классы нового стиля.
33. Наследование. Базовый и производный класс. Построение производного класса.
34. Порождающие функции (функции-фабрики). Множественное наследование. Примеси (Mixin)
35. Агрегация. Контейнеры. Иерархия наследования.
36. Полиморфизм. Подмена методов в производном классе. Доступ к методам базового класса.
37. Специфика разработки программных средств.
38. Жизненный цикл программного средства.
39. Понятие качества программного средства.
40. Обеспечение надежности - основной мотив разработки программного средства. Методы борьбы со сложностью.
41. Обеспечение точности перевода.
42. Преодоление барьера между пользователем и разработчиком.
43. Обеспечение контроля правильности принимаемых решений.
44. Порядок разработки программного модуля.
45. Структурное программирование и пошаговая детализация.
46. Понятие о псевдокоде.
47. Контроль программного модуля.
48. Стратегия проектирования тестов. Заповеди отладки.
49. Автономная отладка и тестирование программного модуля.
50. Комплексная отладка и тестирование программного средства.
51. Документация, создаваемая в процессе разработки программных средств
52. Пользовательская документация программных средств.
53. Документация по сопровождению программных средств.
54. Назначение аттестации программного средства.
55. Испытания и оценка качества программного средства.
56. Виды испытаний и методы оценки качества программного средства.
57. Базовые конструкции языка C++, простые типы данных, структура программы. Операторы языка.
58. Структурные типы данных. Работа с массивами. Алгоритмы сортировки.
59. Строковый тип. Множества. Записи.
60. Подпрограммы
61. Работа с файлами
62. Процедуры и функции модуля Graph
63. Динамическая память и указатели. Динамические структуры данных. Списки.
64. Типы данных. Динамические массивы. Создание консольных приложений в C++.
65. Класс TApplication. Сообщения Windows. События от мыши и клавиатуры.
66. Класс TStringList. Списки. Классификация компонентов. Массив Components.
67. Формы. Компоненты Standard. Класс TCanvas, TGraphic, TPicture.
68. Анимация в C++.
69. Работа с файлами.
70. Динамическое создание компонентов. Создание собственных компонентов.
71. Динамически подключаемые библиотеки.
72. Интерфейсы. Процессы и потоки.
- Примерные тестовые задания для проведения текущего контроля и промежуточной аттестации по итогам освоения дисциплины**
1. Операторы ... являются простой конструкцией условия
 - а) If-Then
 - б) Select Case
 - в) DoWhile
 - г) DoUntil

2. Базовые структуры алгоритма
 - а) следование
 - б) переключатель
 - в) ветвление
 - г) безусловный переход
 - д) цикл
 - е) условный переход
3. Среда разработки программного обеспечения - это ...
 - а) компилятор кода
 - б) система программных средств, используемая для разработки программного обеспечения
 - в) программа, предназначенная для запуска других программ
 - г) программа, предназначенная для написания кода программ
4. Переменная типа Boolean может принимать значения ...
 - а) 0, 1
 - б) True, False
 - в) 1, -1
 - г) -1, 0 1

Типовые контрольные задания

ВАРИАНТ - 1

1. Порядковые типы данных.
2. Составной оператор, оператор выбора.
3. Вычислить: $n!$
4. В двумерном вещественном массиве 3×4 найти строку с наименьшим элементом.
5. Пусть заданы координаты: $x_1, y_1; x_2, y_2; x_3, y_3$ вершин треугольника и координаты x, y любой точки на плоскости. Определить: лежит ли точка с заданными координатами x, y внутри треугольника?

ВАРИАНТ – 2

1. Строки. Процедуры и функции для работы со строками.
2. Условный оператор, оператор цикла с постусловием.
3. Вычислить: $\sin(x) + \sin(xn) + \dots + \sin(xn)$;
4. В двумерном целочисленном массиве 3×4 найти сумму элементов столбца с максимальным элементом.
5. Проверить корректность расстановки скобок в арифметическом выражении.

ВАРИАНТ – 3

1. Множества. Описание и операции над множествами.
2. Оператор безусловного перехода. Оператор цикла с предусловием.
3. Вычислить: $A \cdot (A+1) \cdot (A+2) \cdot \dots \cdot (A+n)$;
4. В заданной строке найти количество слов. Слова могут быть разделены несколькими пробелами.
5. По некоторому каналу связи передается сообщение, имеющее вид последовательности нулей и единиц. Из-за помех возможен ошибочный прием некоторых сигналов: нуль может быть воспринят как единица и наоборот. Для повышения вероятности правильного приема сигналов было решено передавать каждый сигнал трижды. Теперь передатчик вместо 1 всегда передает 111, а вместо 0 всегда 000. Вам предлагается написать программу, которая будет восстанавливать исходное сообщение. При передаче могли произойти ошибки, поэтому вместо каждой тройки цифр программа должна вывести ту цифру, которая встречается в этой тройке по крайней мере два раза.

ВАРИАНТ – 4

1. Тип запись. Тип массив.
2. Оператор цикла for.
3. Вычислить: $\cos(x) + \cos(2 \cdot x) + \dots + \cos(n \cdot x)$;
4. В двумерном целочисленном массиве 3×4 найти сумму элементов главной диагонали.
5. Даны координаты центров столбов, вбитых в вершины выпуклого многоугольника. Известен диаметр столбов. Найти длину натянутой вокруг столбов нити.

7.2. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

а) Критерии оценивания компетенций (результатов).

Программой дисциплины в целях проверки прочности усвоения материала предусматривается проведение различных форм контроля:

1. Текущий контроль - это проверка полноты знаний по основному материалу дисциплинарного модуля (ДМ).

2. Промежуточный контроль - итоговая проверка уровня знаний студента по данной дисциплине в конце семестра (в форме устного или письменного экзамена, сетевого компьютерного тестирования.) Промежуточной формой контроля знаний, умений и навыков по дисциплине является экзамен.

Общий результат выводится как интегральная оценка, складывающаяся из текущего контроля - 50% и промежуточного контроля - 50%.

Текущий контроль по дисциплине включает:

- посещение занятий - 10 баллов,
- участие на практических занятиях (устный опрос, решение задач) - 25 баллов,
- выполнение лабораторных заданий- 25 баллов,
- выполнение домашних контрольных работ (самостоятельная работа) - 10 баллов.

Текущий контроль по ДМ:

письменная контрольная работа -15 баллов;

тестирование - 15 баллов;

Промежуточный контроль по дисциплине включает:

- устный экзамен (тестирование) - 30 баллов,

Критерии оценки посещения занятий- оценка выставляется по 100 бальной системе и соответствует проценту занятий, которые посетил студент из всего количества аудиторных занятий предусмотренных ДМ.

Критерии оценки участия на практических занятиях

Устный опрос. Развернутый ответ студента должен представлять собой связное, логически последовательное сообщение на заданную тему, показывать его умение применять определения, правила в конкретных случаях.

Показатели оценивания:

- 1) полнота и правильность ответа;
- 2) степень осознанности, понимания изученного;
- 3) языковое оформление ответа.

Критерии оценивания устного опроса:

86-100 баллов ставится, если студент полно излагает материал (отвечает на вопрос), дает правильное определение основных понятий; обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только из учебника, но и самостоятельно составленные; излагает материал последовательно и правильно с точки зрения норм литературного языка.

66-85 баллов ставится, если студент дает ответ, удовлетворяющий тем же требованиям, что и для 86-100 баллов, но допускает 1-2 ошибки, которые сам же исправляет, и 1-2 недочета в последовательности и языковом оформлении излагаемого.

51-65 балл ставится, если студент обнаруживает знание и понимание основных положений данной темы, но излагает материал не полно и допускает неточности в определении понятий или формулировке правил; не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.

0-50 баллов ставится, если студент обнаруживает незнание большей части соответствующего вопроса, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал.

Решение задач.

86-100 баллов выставляется, если студент демонстрирует знания теоретического и практического материала по теме практической работы, определяет взаимосвязи между показателями задачи, даёт правильный алгоритм решения, определяет междисциплинарные связи по условию задания.

66-85 баллов выставляется, если студент демонстрирует знания теоретического и практического материала по теме практической работы, допуская незначительные неточности при решении задач, имея неполное понимание междисциплинарных связей при правильном выборе

алгоритма решения задания.

51-65 балл выставляется, если студент затрудняется с правильной оценкой предложенной задачи, даёт неполный ответ, требующий наводящих вопросов преподавателя, выбор алгоритма решения задачи возможен при наводящих вопросах преподавателя.

0-50 баллов выставляется студенту, если он даёт неверную оценку ситуации, неправильно выбирает алгоритм решения.

Критерии оценки выполнения лабораторных заданий.

Основными показателями оценки выполненной студентом и представленной для проверки работы являются:

1. Степень соответствия выполненного задания поставленным целям, задачам и требованиям;
2. Оформление, структурирование и комментирование лабораторной работы;
3. Уникальность выполнения работы (отличие от работ коллег);
4. Успешные ответы на контрольные вопросы.

Критерии оценки лабораторной работы.

86-100 баллов - оформление соответствует требованиям, критерии выдержаны, защита всего перечня контрольных вопросов.

66-85 баллов - оформление соответствует требованиям, критерии выдержаны, защита только до 85 % контрольных вопросов.

51-65 балл - оформление соответствует требованиям, критерии выдержаны, защита только до 51 % контрольных вопросов.

0-50 баллов - оформление не соответствует требованиям, критерии не выдержаны, защита только менее 51 % контрольных вопросов.

Критерии оценки выполнения домашних контрольных работ (самостоятельная работа).

Основными показателями оценки выполненной студентом и представленной для проверки домашней контрольной работы являются:

1. Степень соответствия выполненного задания поставленным целям, задачам и требованиям;
2. Оформление, структурирование и комментирование лабораторной работы;
3. Уникальность выполнения работы (отличие от работ коллег);
4. Успешные ответы на контрольные вопросы.

Критерии оценки домашней контрольной работы.

86-100 баллов - студент правильно выполнил индивидуальное самостоятельное задание. Показал отличные владения навыками применения полученных знаний и умений при решении профессиональных задач в рамках усвоенного учебного материала. Ответил на все дополнительные вопросы на защите.

66-85 баллов - студент выполнил индивидуальное самостоятельное задание с небольшими неточностями. Показал хорошие владения навыками применения полученных знаний и умений при решении профессиональных задач в рамках усвоенного учебного материала. Ответил на большинство дополнительных вопросов на защите.

51-65 балл - студент выполнил индивидуальное самостоятельное задание с существенными неточностями. Показал удовлетворительное владение навыками применения полученных знаний и умений при решении профессиональных задач в рамках усвоенного учебного материала. При ответах на дополнительные вопросы на защите было допущено много неточностей.

0-50 баллов - при выполнении индивидуального самостоятельного задания студент продемонстрировал недостаточный уровень владения умениями и навыками при решении профессиональных задач в рамках усвоенного учебного материала. При ответах на дополнительные вопросы на защите было допущено множество неточностей.

Критерии оценки текущего контроля по ДМ (письменная контрольная работа и тестирование).

Письменная контрольная работа состоит из двух типов вопросов:

1. Теоретические вопросы из курса лекций и практических работ. - 40 баллов.
2. Практические вопросы и задачи по лекционному и практическому материалу. - 60 баллов.

86-100 баллов - студент, показал всестороннее, систематическое и глубокое знание учебного материала, самостоятельно ответил на вопросы, ответ отличается богатством и точностью использованных терминов, материал излагается последовательно и логично; показал отличные владения навыками применения полученных знаний и умений при решении профессиональных задач.

66-85 баллов - студент, показал полное знание учебного материала, не допускающий в ответе существенных неточностей, самостоятельно выполнивший ответивший на вопросы; показал хорошие владения навыками применения полученных знаний и умений при решении профессиональных задач

51-65 балл - студент, обнаруживший знание основного учебного материала в объёме, необходимом для дальнейшей учебы самостоятельно выполнивший задания, однако допустивший некоторые погрешности при ответе на вопросы; показал удовлетворительное владение навыками применения полученных знаний и умений при решении профессиональных задач.

0-50 баллов - выставляется студенту, обнаружившему пробелы в знаниях или отсутствие знаний по значительной части основного учебного материала, не выполнившего задания, допустившему принципиальные ошибки при ответе на вопросы, продемонстрировал недостаточный уровень владения умениями и навыками при решении профессиональных задач.

Критерии выставления оценок за *тестирование* Тестовое задание состоит из пятнадцати вопросов. Время выполнения работы: 15-20 мин.

86-100 баллов - оценка «отлично» - 13-15 правильных ответов;

66-85 баллов - оценка «хорошо» - 10-12 правильных ответов;

51-65 балл - оценка «удовлетворительно» - 8-9 правильных ответов;

0-50 баллов - оценка «неудовлетворительно» - менее 8 правильных ответов.

Критерии оценки устного экзамена

Экзаменационные билеты включают три типа заданий:

1. Теоретические вопросы из курса лекций и практических работ. - 30 баллов.

2. Практические вопросы по лекционному и практическому материалу. - 40 баллов.

3. Проблемные вопросы и расчетные задачи. - 40 баллов.

Проверка качества подготовки студентов на экзаменах заканчивается выставлением отметок по принятой пятибалльной шкале («отлично», «хорошо», «удовлетворительно», «неудовлетворительно»).

86-100 баллов - оценка «отлично» - студент владеет знаниями по дисциплине **«Технологии и методы программирования»** в полном объеме учебной программы, достаточно глубоко осмысливает дисциплину; самостоятельно, в логической последовательности и исчерпывающе отвечает на все вопросы билета, подчеркивая при этом самое существенное, умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделять в нем главное; устанавливать причинно-следственные связи; четко формирует ответы, свободно решает ситуационные задачи повышенной сложности; хорошо знаком с основной литературой; увязывает теоретические аспекты дисциплины с прикладными задачами исследования операций и методов оптимизации; владеет современными информационными технологиями решения прикладных задач.

66-85 баллов - оценка «хорошо» - студент владеет знаниями дисциплины **«Технологии и методы программирования»** 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 легкие и средней тяжести ситуационные задачи; умеет трактовать выбор тех или иных методов и средств решения прикладных задач.

51-65 балл - оценка «удовлетворительно» - студент владеет основным объемом знаний по дисциплине **«Технологии и методы программирования»**; проявляет затруднения в самостоятельных ответах, оперирует неточными формулировками; в процессе ответов допускаются ошибки по существу вопросов. Студент способен решать лишь наиболее легкие задачи, владеет только обязательным минимумом при решении задач исследования операций.

0-50 баллов - оценка «неудовлетворительно» - студент не освоил обязательного минимума знаний дисциплины **«Технологии и методы программирования»**, не способен ответить на вопросы билета даже при дополнительных наводящих вопросах экзаменатора.

Критерии оценки экзамена в форме тестирования

Тестовое задание состоит из тридцати вопросов. Время выполнения работы: 60 мин.

86-100 баллов - оценка «отлично» - 26-30 правильных ответов;

66-85 баллов - оценка «хорошо» - 20-25 правильных ответов;

51-65 балл - оценка «удовлетворительно» - 16-19 правильных ответов;

0-50 баллов - оценка «неудовлетворительно» - менее 16 правильных ответов.

8. Учебно-методическое обеспечение дисциплины.

а) основная литература:

1. Мейер Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс]/ Мейер Б.- Электрон. текстовые данные.- М.: Интернет Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019.- 285 с.- Режим доступа: <http://www.iprbookshop.ru/79706.html> . – ЭБС «IPRbooks».

2. Биллиг В.А. Основы программирования на C# [Электронный ресурс]/ Биллиг В.А.- Электрон. текстовые данные.- М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.- 574 с.- Режим доступа: <http://www.iprbookshop.ru/73695.html> .- ЭБС «IPRbooks».

3. Роганов Е.А. Основы информатики и программирования [Электронный ресурс] / Е.А. Роганов. - 2-е изд. - Электрон.текстовые данные. - М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 392 с. - 2227-8397. - Режим доступа: <http://www.iprbookshop.ru/73689.html> (дата обращения: 07.06.2022)

б) дополнительная литература:

4. Васильев А.Н. Python на примерах [Электронный ресурс]: практический курс по программированию/ Васильев А.Н.- Электрон. текстовые данные- Санкт-Петербург: Наука и Техника, 2017.- 432 с.- Режим доступа: <http://www.iprbookshop.ru/73043.html>.- ЭБС «IPRbooks»

5. Сузи Р.А. Язык программирования Python [Электронный ресурс]: учебное пособие/ Сузи Р.А.- Электрон. текстовые данные- Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020.- 350 с.- Режим доступа: <http://www.iprbookshop.ru/97589.html>.- ЭБС «IPRbooks»

6. Информатика [Электронный ресурс]: учебное пособие. - Электрон.текстовые данные. - Ставрополь: Северо-Кавказский федеральный университет, 2016. - 178 с. - 2227-8397. - Режим доступа: <http://www.iprbookshop.ru/66024.html> (дата обращения: 07.06.2022)

7. Никифоров С.Н. Информатика. Часть 3. Прикладное программирование [Электронный ресурс]: учебное пособие / С.Н. Никифоров. - Электрон.текстовые данные. - СПб.: Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2016. - 128 с. - 978-5-9227-0743-5. - Режим доступа: <http://www.iprbookshop.ru/74384.html> (дата обращения: 07.06.2022)

8. Прохорова О.В. Информатика [Электронный ресурс]: учебник / О.В. Прохорова. - Электрон.текстовые данные. - Самара: Самарский государственный архитектурно-строительный университет, ЭБС АСВ, 2013. - 106 с. - 978-5-9585-0539-5. - Режим доступа: <http://www.iprbookshop.ru/20465.html> (дата обращения: 07.06.2022)

9. Шелудько В.М. Основы программирования на языке высокого уровня Python [Электронный ресурс]: учебное пособие/ Шелудько В.М.- Электрон.текстовые данные- Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017.- 146 с.- Режим доступа: <http://www.iprbookshop.ru/87461.html>.- ЭБС «IPRbooks»

10. Яшин В.Н. Информатика: аппаратные средства персонального компьютера: учеб.пособие / Яшин, Владимир Николаевич. - М.: ИНФРА-М, 2010. - 252,[4] с. - (Высшее образование: серия основана в 1996 г.). - Допущено УМО. - ISBN 978-5-16-003190-3: 153-23. (Количество экз. - 10)

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Портал доступа к электронным образовательным ресурсам ДГУ [Электронный ресурс] Режим доступа URL: <http://dgu.ru> (дата обращения: 1.07.2022)

2. Электронно-библиотечная система IPRbooks [Электронный ресурс] Режим доступа URL: <http://www.iprbookshop.ru> (дата обращения: 1.07.2022)

3. Электронно-библиотечная система «Университетская библиотека онлайн» [Электронный ресурс] Режим доступа URL: <http://biblioclub.ru> (дата обращения: 1.09.2022)

4. Электронно-библиотечная система издательства «Инфра» [Электронный ресурс] Режим доступа URL: <http://znanium.com> (дата обращения: 1.09.2022)

5. IT-портал [Электронный ресурс] Режим доступа URL: <http://citforum.ru>

6. Портал Национального открытого университета «Интуит» [Электронный ресурс] Режим доступа URL: <http://www.intuit.ru> (дата обращения: 1.09.2022)

7. Техническая документация фирмы Microsoft [Электронный ресурс] Режим доступа URL: <http://technet.microsoft.com/ru-ru/sysinternals>(дата обращения: 1.07.2022)

8. Moodle[Электронный ресурс]: система виртуального обучением: [база данных] / Даг.гос. ун-т. - Махачкала, г. - Доступ из сети ДГУ или, после регистрации из сети ун-та, из любой точки, имеющей доступ в интернет. -Режим доступа URL: <http://moodle.dgu.ru/>(дата обращения: 1.09.2022).

9. Учебный курс по программированию «Учите Питон» [Электронный ресурс] - Режим доступа: <http://pythontutor.ru/> (дата обращения: 1.07.2022)

10.Платформа учебных курсов Stepik [Электронный ресурс] - Режим доступа URL: <https://stepik.org/catalog?q=Python> (дата обращения: 1.07.2022)

10. Методические указания для обучающихся по освоению дисциплины

Для изучения теоретического курса студентам необходимо использовать лекционный материал, учебники и учебные пособия из списка основной и дополнительной литературы, интернет источники.

По дисциплине «Технологии и методы программирования» в конце каждого модуля проводится контрольная работа.

В контрольную работу включаются теоретические вопросы и задачи тех типов, которые были разобраны на предшествующих практических занятиях.

Рабочей программой дисциплины «Технологии и методы программирования» предусмотрена самостоятельная работа студентов в объеме 144 часа. Самостоятельная работа проводится с целью углубления знаний по дисциплине и предусматривает:

- чтение студентами рекомендованной литературы и усвоение теоретического материала дисциплины;

- подготовку к практическим занятиям;

- выполнение индивидуальных заданий;

- подготовку к контрольным работам, зачету и экзаменам.

С самого начала изучения дисциплины студент должен четко уяснить, что без систематической самостоятельной работы успех невозможен. Эта работа должна регулярно начинаться сразу после лекционных и практических занятий, для закрепления только что пройденного материала.

После усвоения теоретического материала можно приступить к самостоятельному решению задач из учебников и пособий, входящих в список основной литературы.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем.

Интернет-ресурсы, мультимедиа, электронная почта для коммуникации со студентами.

Обучающие интерактивные интернет-порталы, интернет-ресурсы, мультимедиа, электронная почта для коммуникации со студентами, системы программирования, MicrosoftOffice.

Справочно-правовые системы

1. Компьютерные классы с доступом к сети интернет и с набором лицензионного базового программного обеспечения для проведения лабораторных занятий;

2. Python IDE (PyCharm), C++ для выполнения лабораторных заданий

3. Лекционная мультимедийная аудитория для чтения лекций с использованием мультимедийных материалов.

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Компьютерные классы и лаборатории, оборудованные для проведения лекционных, практических и лабораторных занятий средствами оргтехники, персональными компьютерами, объединенными в сеть с выходом в Интернет; установленное лицензионное и свободное программное обеспечение.

Для выполнения лабораторных работ используется компьютерное оборудование с установленными программными продуктами MSOffice, Интерпретатор Python, C++, Notebook. Аудиторные занятия проводятся в компьютерных классах с доступом к сети Интернет.