

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Горно-Алтайский государственный университет»
(ФГБОУ ВО ГАГУ, ГАГУ, Горно-Алтайский государственный университет)

Технологии программирования рабочая программа дисциплины (модуля)

Закреплена за кафедрой **кафедра математики, физики и информатики**

Учебный план 02.03.01_2025_625.plx
02.03.01 Математика и компьютерные науки
Цифровые технологии

Квалификация **бакалавр**

Форма обучения **очная**

Общая трудоемкость **19 ЗЕТ**

Часов по учебному плану	684	Виды контроля в семестрах:
в том числе:		экзамены 4, 6, 5, 3
аудиторные занятия	306	зачеты с оценкой 2
самостоятельная работа	219,6	
часов на контроль	147,85	

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	2 (1.2)		3 (2.1)		4 (2.2)		5 (3.1)		6 (3.2)		
Неделя	17 2/6		17 1/6		16 3/6		16 1/6		15 5/6		
Вид занятий	УП	РП	УП	РП	УП	РП	УП	РП	УП	РП	
Лекции	18	18	18	18	36	36	18	18	18	18	1
Лабораторные	36	36	36	36	54	54	36	36	36	36	1
Консультации (для студента)	0,9	0,9	0,9	0,9	1,8	1,8	0,9	0,9	0,9	0,9	5
Контроль самостоятельной работы при проведении аттестации	0,15	0,15	0,25	0,25	0,25	0,25	0,25	0,25	0,25	0,25	1
Консультации перед экзаменом			1	1	1	1	1	1	1	1	4
Итого ауд.	54	54	54	54	90	90	54	54	54	54	3
Контактная работа	55,05	55,05	56,15	56,15	93,05	93,05	56,15	56,15	56,15	56,15	3
Сам. работа	44,1	44,1	17,1	17,1	88,2	88,2	17,1	17,1	53,1	53,1	2
Часы на контроль	8,85	8,85	34,75	34,75	34,75	34,75	34,75	34,75	34,75	34,75	1
Итого	108	108	108	108	216	216	108	108	144	144	6

Программу составил(и):

ст.преподаватель, Кречетова Марина Юрьевна

Рабочая программа дисциплины

Технологии программирования

разработана в соответствии с ФГОС:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 02.03.01 Математика и компьютерные науки (приказ Минобрнауки России от 23.08.2017 г. № 807)

составлена на основании учебного плана:

02.03.01 Математика и компьютерные науки

утвержденного учёным советом вуза от 30.01.2025 протокол № 2.

Рабочая программа утверждена на заседании кафедры

кафедра математики, физики и информатики

Протокол от 10.04.2025 протокол № 10

Зав. кафедрой Богданова Рада Александровна

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2026-2027 учебном году на заседании кафедры
кафедра математики, физики и информатики

Протокол от _____ 2026 г. № ____
Зав. кафедрой Богданова Рада Александровна

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2027-2028 учебном году на заседании кафедры
кафедра математики, физики и информатики

Протокол от _____ 2027 г. № ____
Зав. кафедрой Богданова Рада Александровна

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2028-2029 учебном году на заседании кафедры
кафедра математики, физики и информатики

Протокол от _____ 2028 г. № ____
Зав. кафедрой Богданова Рада Александровна

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2029-2030 учебном году на заседании кафедры
кафедра математики, физики и информатики

Протокол от _____ 2029 г. № ____
Зав. кафедрой Богданова Рада Александровна

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1	<i>Цели:</i> Обучение методам проектирования, описания на языке программирования высокого уровня и тестирования алгоритмов решения математических задач.
1.2	<i>Задачи:</i> 1. Изучение основ программирования на языке высокого уровня. 2. Формирование знаний, умений и навыков, позволяющих овладеть методами конструирования, тестирования и анализа алгоритмов и их реализаций на компьютере. 3. Формирование навыков решения математических задач на компьютере.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Цикл (раздел) ООП:		Б1.О.16
2.1	Требования к предварительной подготовке обучающегося:	
2.1.1	Программа предназначена для студентов 1-3 курсов. На начальном этапе изучения дисциплины от студентов требуется владение базовыми математическими понятиями, достаточными для работы с формулировками математических и алгоритмических утверждений.	
2.2	Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:	
2.2.1	Цифровые технологии в профессиональной деятельности	
2.2.2	Проектная деятельность в цифровых технологиях	
2.2.3	Базы данных	
2.2.4	Операционные системы	
2.2.5	Компьютерные сети	
2.2.6	Архитектура ЭВМ	
2.2.7	Web-технологии	
2.2.8	Введение в машинное обучение и искусственный интеллект	
2.2.9	Методика обучения информатике и ИКТ в школе	
2.2.10	Нейронные сети	
2.2.11	Анализ данных в R	
2.2.12	Основы программирования в R	
2.2.13	Получение первичных навыков разработки и представления IT-проектов	
2.2.14	Технологическая (проектно-технологическая) практика	
2.2.15	Получение навыков разработки web-приложений	

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

ОПК-6: Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	
ИД-1.ОПК-6: Знает основные языки программирования и принципы работы с базами данных	
- знать основные подходы к разработке алгоритмов и программ;	
ИД-2.ОПК-6: Умеет применять языки программирования, технологии баз данных для решения практических задач	
- уметь применять методы программирования для решения типовых математических и простых прикладных задач информатики;	
ИД-3.ОПК-6: Владеет навыками программирования, отладки и тестирования программ	
- владеть математическими и компьютерными методами решения типовых задач;	
ПК-2: Способен искать нужные источники информации и данные, воспринимать, анализировать, запоминать и обрабатывать программный код информации с использованием цифровых средств, а также с помощью алгоритмов и программ с целью эффективного использования информации для решения профессиональных задач с учетом основных требований информационной безопасности	
ИД-3.ПК-2: Способен анализировать обрабатывать программный код информации с использованием алгоритмов и прикладных программ	
умеет анализировать базовые алгоритмы и прикладные программы	

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)							
Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетен-ции	Литература	Инте ракт.	Примечание
	Раздел 1. Лекции 2 семестр						
1.1	Язык C++. Типы данных. Ветвления и циклы. /Лек/	2	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
1.2	Язык C++. Векторы и строки. /Лек/	2	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
1.3	Язык C++. Составные типы данных. /Лек/	2	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
1.4	Язык C++. Ссылки, указатели, константность. /Лек/	2	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
1.5	Язык C++. Функции. /Лек/	2	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
1.6	Язык C++. Шаблоны. /Лек/	2	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 2. Лабораторные работы 2 семестр						
2.1	Язык C++. Типы данных. Ветвления и циклы. /Лаб/	2	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
2.2	Язык C++. Векторы и строки. /Лаб/	2	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
2.3	Язык C++. Составные типы данных. /Лаб/	2	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
2.4	Язык C++. Ссылки, указатели, константность. /Лаб/	2	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
2.5	Язык C++. Функции. /Лаб/	2	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
2.6	Язык C++. Шаблоны. /Лаб/	2	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 3. Самостоятельная работа 2 семестр						
3.1	Язык C++. Типы данных. Ветвления и циклы. /Ср/	2	7		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	

3.2	Язык C++. Векторы и строки. /Ср/	2	7		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
3.3	Язык C++. Составные типы данных. /Ср/	2	7		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
3.4	Язык C++. Ссылки, указатели, константность. /Ср/	2	7		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
3.5	Язык C++. Функции. /Ср/	2	7		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
3.6	Язык C++. Шаблоны. /Ср/	2	9,1		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
Раздел 4. Консультации							
4.1	Консультация по дисциплине /Конс/	2	0,9	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
Раздел 5. Промежуточная аттестация (зачёт)							
5.1	Подготовка к зачёту /ЗачётСОц/	2	8,85	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
5.2	Контактная работа /КСРАтт/	2	0,15	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
Раздел 6. Лекции 3 семестр							
6.1	Язык C++. Последовательные контейнеры. /Лек/	3	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
6.2	Язык C++. Ассоциативные контейнеры. /Лек/	3	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
6.3	Язык C++. Классы. /Лек/	3	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
6.4	Язык C++. Наследование и полиморфизм. /Лек/	3	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
6.5	Язык C++. Обработка исключений. /Лек/	3	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	

6.6	Язык C++. Идиома RAII и умные указатели. /Лек/	3	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 7. Лабораторные работы 3 семестр						
7.1	Язык C++. Последовательные контейнеры. /Лаб/	3	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
7.2	Язык C++. Ассоциативные контейнеры. /Лаб/	3	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
7.3	Язык C++. Классы. /Лаб/	3	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
7.4	Язык C++. Наследование и полиморфизм. /Лаб/	3	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
7.5	Язык C++. Обработка исключений /Лаб/	3	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
7.6	Язык C++. Идиома RAII и умные указатели. /Лаб/	3	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 8. Самостоятельная работа 3 семестр						
8.1	Язык C++. Последовательные контейнеры. /Ср/	3	2,5		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
8.2	Язык C++. Ассоциативные контейнеры. /Ср/	3	2,5		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
8.3	Язык C++. Классы. /Ср/	3	2,5		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
8.4	Язык C++. Наследование и полиморфизм. /Ср/	3	2,5		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
8.5	Язык C++. Обработка исключений. /Ср/	3	2,5		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
8.6	Язык C++. Идиома RAII и умные указатели. /Ср/	3	4,6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 9. Консультации						
9.1	Консультация по дисциплине /Конс/	3	0,9	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	

	Раздел 10. Промежуточная аттестация (экзамен)						
10.1	Подготовка к экзамену /Экзамен/	3	34,75	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
10.2	Контроль СР /КСРАтт/	3	0,25	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
10.3	Контактная работа /КонсЭк/	3	1	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 11. Лекции 4 семестр						
11.1	Язык Python. Базовые конструкции Python. /Лек/	4	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
11.2	Язык Python. Строки, кортежи, списки. /Лек/	4	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
11.3	Язык Python. Множества, словари. /Лек/	4	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
11.4	Язык Python. Списочные выражения. Модель памяти для типов языка Python. /Лек/	4	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
11.5	Язык Python. Поточковый ввод/вывод. Работа с текстовыми файлами. JSON. /Лек/	4	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
11.6	Язык Python. Функции и их особенности в Python. /Лек/	4	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 12. Лабораторные работы 4 семестр						
12.1	Язык Python. Базовые конструкции Python. /Лаб/	4	9		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
12.2	Язык Python. Строки, кортежи, списки. /Лаб/	4	9		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
12.3	Язык Python. Множества, словари. /Лаб/	4	9		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
12.4	Язык Python. Списочные выражения. Модель памяти для типов языка Python. /Лаб/	4	9		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	

12.5	Язык Python. Поточковый ввод/вывод. Работа с текстовыми файлами. JSON. /Лаб/	4	9		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
12.6	Язык Python. Функции и их особенности в Python. /Лаб/	4	9		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 13. Самостоятельная работа 4 семестр						
13.1	Язык Python. Базовые конструкции Python. /Ср/	4	15		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
13.2	Язык Python. Строки, кортежи, списки. /Ср/	4	15		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
13.3	Язык Python. Множества, словари. /Ср/	4	15		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
13.4	Язык Python. Списочные выражения. Модель памяти для типов языка Python. /Ср/	4	15		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
13.5	Язык Python. Поточковый ввод/вывод. Работа с текстовыми файлами. JSON. /Ср/	4	15		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
13.6	Язык Python. Функции и их особенности в Python. /Ср/	4	13,2		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 14. Консультации						
14.1	Консультация по дисциплине /Конс/	4	1,8	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 15. Промежуточная аттестация (экзамен)						
15.1	Подготовка к экзамену /Экзамен/	4	34,75	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
15.2	Контроль СР /КСРАтт/	4	0,25	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
15.3	Контактная работа /КонсЭк/	4	1	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 16. Лекции 5 семестр						

16.1	Разработка пользовательского интерфейса доступа к БД. /Лек/	5	18		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 17. Лабораторные работы 5 семестр						
17.1	Разработка пользовательского интерфейса доступа к БД. /Лаб/	5	36		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 18. Самостоятельная работа 5 семестр						
18.1	Разработка пользовательского интерфейса доступа к БД. /Ср/	5	17,1		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 19. Консультации						
19.1	Консультация по дисциплине /Конс/	5	0,9	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 20. Промежуточная аттестация (экзамен)						
20.1	Подготовка к экзамену /Экзамен/	5	34,75	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
20.2	Контроль СР /КСРАтт/	5	0,25	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
20.3	Контактная работа /КонсЭк/	5	1	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 21. Лекции 6 семестр						
21.1	Язык Python. Объектная модель Python. Классы, поля и методы. /Лек/	6	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
21.2	Язык Python. Волшебные методы, переопределение методов. Наследование. /Лек/	6	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
21.3	Язык Python. Модель исключений Python. Try, except, else, finally. Модули. /Лек/	6	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
21.4	Язык Python. Модули math и numpy. /Лек/	6	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
21.5	Язык Python. Модуль pandas. /Лек/	6	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	

21.6	Язык Python. Модуль Matplotlib. /Лек/	6	3		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 22. Лабораторные работы 6 семестр						
22.1	Язык Python. Объектная модель Python. Классы, поля и методы. /Лаб/	6	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
22.2	Язык Python. Волшебные методы, переопределение методов. Наследование. /Лаб/	6	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
22.3	Язык Python. Модель исключений Python. Try, except, else, finally. Модули. /Лаб/	6	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
22.4	Язык Python. Модули math и numpy. /Лаб/	6	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
22.5	Язык Python. Модуль pandas. /Лаб/	6	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
22.6	Язык Python. Модуль Matplotlib. /Лаб/	6	6		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 23. Самостоятельная работа 6 семестр						
23.1	Язык Python. Объектная модель Python. Классы, поля и методы. /Ср/	6	9		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
23.2	Язык Python. Волшебные методы, переопределение методов. Наследование. /Ср/	6	9		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
23.3	Язык Python. Модель исключений Python. Try, except, else, finally. Модули. /Ср/	6	9		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
23.4	Язык Python. Модули math и numpy. /Ср/	6	9		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
23.5	Язык Python. Модуль pandas. /Ср/	6	9		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
23.6	Язык Python. Модуль Matplotlib. /Ср/	6	8,1		Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
	Раздел 24. Консультации						
24.1	Консультация по дисциплине /Конс/	6	0,9	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	

	Раздел 25. Промежуточная аттестация (экзамен)						
25.1	Подготовка к экзамену /Экзамен/	6	34,75	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
25.2	Контроль СР /КСРАтт/	6	0,25	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	
25.3	Контактная работа /КонсЭк/	6	1	ИД-1.ОПК-6 ИД-2.ОПК-6 ИД-3.ОПК-6 ИД-3.ПК-2	Л1.1 Л1.2 Л1.3 Л1.4 Л1.5Л2.1 Л2.2 Л2.3	0	

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Пояснительная записка

1. Назначение фонда оценочных средств. Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «Технологии программирования».
2. Фонд оценочных средств включает примерный тест для проведения входного контроля, примеры задач для лабораторных работ, примерный перечень вопросов к экзамену и зачету, примерные темы курсовых работ.

5.2. Оценочные средства для текущего контроля

Примерный тест для входного контроля

1. Что называется алгоритмом?
 1. последовательность команд, которую может выполнить исполнитель
 2. система команд исполнителя
 3. нумерованная последовательность строк
 4. ненумерованная последовательность строк
2. Что такое исполнитель алгоритма?
 1. Это список команд для решения поставленной задачи.
 2. Это программа, составленная по заданному алгоритму.
 3. Это объект, который способен понимать и исполнять команды, указанные в алгоритме.
3. Какой алгоритм называется циклическим?
 1. Алгоритм, в котором команды работают последовательно одна за другой.
 2. Алгоритм, в котором команда или несколько команд работают многократно.
 3. Алгоритм, который работает либо по одной ветви, либо по другой, в зависимости от выполнения условия.
4. Какой алгоритм называется линейным?
 1. Алгоритм, в котором команды работают последовательно одна за другой.
 2. Алгоритм, в котором команда или несколько команд работают многократно.
 3. Алгоритм, который работает либо по одной ветви, либо по другой, в зависимости от выполнения условия.
5. Какой алгоритм называется алгоритмом ветвления?
 1. Алгоритм, в котором команды работают последовательно одна за другой.
 2. Алгоритм, в котором команда или несколько команд работают многократно.
 3. Алгоритм, который работает либо по одной ветви, либо по другой, в зависимости от выполнения условия.
6. Модель есть замещение изучаемого объекта другим объектом, который отражает...
 1. все стороны данного объекта
 2. некоторые стороны данного объекта
 3. существенные стороны данного объекта
 4. несущественные стороны данного объекта
7. Модель содержит информации...

1. столько же, сколько и моделируемый объект
2. меньше, чем моделируемый объект
3. больше, чем моделируемый объект
4. не содержит информации

8. Каковы основные этапы обработки информации компьютером?

1. Ввод и вывод информации.
2. Ввод, преобразование, хранение, вывод информации.
3. Сохранение информации в файле.

9. Какой этап решения задачи на компьютере отсутствует в следующей цепочке: объект - ... - исследование модели на компьютере - анализ результатов и корректировка модели?

1. построение информационной модели
2. кодировка алгоритма на языке программирования
3. анализ полученных данных
4. разработка алгоритма

10. Свойством алгоритма является:

1. Результативность;
2. Цикличность;
3. Возможность изменения последовательности выполнения команд;
4. Возможность выполнения алгоритма в обратном порядке;
5. Простота записи на языках программирования.

Критерии оценки

«Зачтено» – выполнение верно более 60% заданий.

«Не зачтено» – выполнение 60% и менее заданий верно.

Текущий контроль осуществляется по выполнению практических работ, результаты которых проверяются в конце занятия.

Примерные задачи для лабораторных работ

Основы языка C++

Управляющие конструкции:

1. Услуги телефонной сети оплачиваются по следующему правилу: за разговоры до А минут в месяц — В руб., а разговоры сверх установленной нормы оплачиваются из расчета С руб. за минуту. Написать программу, вычисляющую плату за пользование телефоном для введенного времени разговоров за месяц.

2. Напишите программу, которая по заданной оценке за успеваемость от 2 до 5 выводит, сообщение «неудовлетворительно», «удовлетворительно», «хорошо» или «отлично».

3. Напишите программу определения количества корней квадратного уравнения, где коэффициенты уравнения задаются с клавиатуры.

4. С клавиатуры вводится последовательность чисел до тех пор, пока не будет задан ноль, который элементом не считается. Определите количество положительных, отрицательных чисел.

Структуры:

1. Описать структуру с именем STUDENT, содержащую следующие поля:

- NAME - фамилия и инициалы;
- Номер группы;
- Успеваемость (целое число).

Написать программу, выполняющую следующие действия:

- Ввод с клавиатуры данных в массив, состоящий из четырех структур типа STUDENT.
- Вывод списка структур и среднего балла.

Для массива структур формулировка задания может содержать нахождение максимального, минимального значения какого-либо поля, суммы или количества записей удовлетворяющих какому-либо условию для заданного поля.

Одномерные массивы:

1. Дан целочисленный массив из 30 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на одном из языков программирования алгоритм, который находит количество элементов массива, больших 100 и при этом кратных 5, а затем заменяет каждый такой элемент на число, равное найденному количеству. Гарантируется, что хотя бы один такой элемент в массиве есть. В качестве результата необходимо вывести измененный массив, каждый элемент массива выводится с новой строки.

Например, для массива из шести элементов: 4 115 7 195 25 106 программа должна вывести числа 4 2 7 2 25 106

Для одномерного массива формулировка задания может содержать нахождение суммы, количества элементов, удовлетворяющих заданному условию, а также минимального или максимального элемента массива.

Двумерные массивы:

1. Найдите максимальные элементы столбцов (строк, главной диагонали) матрицы размерностью 5×5 , значения которой задаются случайным образом из диапазона от 0 до 9.
2. Найдите максимальный элемент всей матрицы размерностью 5×5 , значения которой задаются случайным образом из диапазона от 0 до 9. выведите на экран количество максимальных элементов.
3. Найдите сумму (разность, умножение матрицы на заданное число) матриц $A[n, m]$ и $B[n, m]$.
Для двумерного массива формулировка задания может содержать нахождение суммы, количества элементов, удовлетворяющих заданному условию, а также минимального или максимального элемента массива для всей матрицы, для строки, для столбца, главной и побочной диагонали. Также возможно задание на нахождение суммы, разности, произведения (умножение на число) матриц.

Строки:

1. Считать с клавиатуры две строки (в отдельные строковые переменные). Сформировать новую (третью) строку, которая будет содержать введенные строки разделенные пробелом. Написать решение задачи с использованием библиотек `cstring` и `string`.

Примерна формулировка «комплексной» задачи

1. Описать структуру с именем `STUDENT`, содержащую следующие поля:

- `NAME` - фамилия и инициалы;
- Номер группы;
- Успеваемость (целое число).

Написать программу, выполняющую следующие действия:

- Ввод из текстового файла значений в массив, состоящий из четырех структур типа `STUDENT`.
- Вывод списка структур (с помощью функцию вывода значений структуры) и среднего балла.

Основы языка Python**Циклы**

1. Последовательно вводятся числа. Определить сумму чисел с нечётными номерами и произведение чисел с чётными номерами (по порядку ввода). Подсчитать количество слагаемых и количество сомножителей. При вводе числа 55555 закончить работу.

Списки

1. Дан одномерный массив числовых значений, насчитывающий N элементов. Поменять местами элементы, стоящие на чётных и нечётных местах: $A[1] \leftrightarrow A[2]$; $A[3] \leftrightarrow A[4]$.
2. Дан одномерный массив числовых значений, насчитывающий N элементов. Сумму элементов массива и количество положительных элементов поставить на первое и второе место.

Строки

1. Заданы M строк символов, которые вводятся с клавиатуры. Найти количество символов в самой длинной строке. Выровнять строки по самой длинной строке, поставив перед каждой строкой соответствующее количество звёздочек.
2. Заданы M строк символов, которые вводятся с клавиатуры. Из заданных строк, каждая из которых представляет одно слово, составить одну длинную строку, разделяя слова пробелами.

Двумерные массивы

1. Выполнить обработку элементов прямоугольной матрицы A , имеющей N строк и M столбцов. Найти наибольший элемент столбца матрицы A , для которого сумма абсолютных значений элементов максимальна.
2. Выполнить обработку элементов прямоугольной матрицы A , имеющей N строк и M столбцов. Найти наибольшее значение среди средних значений для каждой строки матрицы.

Типовые задачи на кортежи и множества

1. Дан список чисел, который может содержать до 100000 чисел. Определите, сколько в нем встречается различных чисел.
2. Во входном файле (вы можете читать данные из файла `input.txt`) записан текст. Словом считается последовательность непробельных символов идущих подряд, слова разделены одним или большим числом пробелов или символами конца строки.
Определите, сколько различных слов содержится в этом тексте.

Типовые задачи на разработку пользовательского интерфейса доступа к БД.

1. Изучить `SQLite3`. Умения: Создание базы данных. Создание таблицы. Запрос на добавление данных в таблицу. Запрос на выборку данных из таблицы. Запрос на удаление данных. Отображение структуры базы данных (перечень таблиц). Отображение содержимого таблицы.
2. Создадим базу данных «Фирма» из трех таблиц: Сотрудники, Клиенты и Заказы. Организуйте связи между таблицами. Заполните таблицы БД не менее чем 10-ю записями. Осуществите отбор данных с помощью запросов различных типов. Создайте для работы с каждой таблицей формы и одну кнопочную форму, с которой осуществляется открытие форм для отдельных таблиц. Создайте отчеты на основе данных таблиц и запросов.

Типовые задачи по основам ООП

1. Разработайте класс `Point`, который при инициализации принимает координаты точки на декартовой плоскости и сохраняет их в поля `x` и `y` соответственно. Реализуйте методы:

`move`, который перемещает точку на заданное расстояние по осям `x` и `y`;

`length`, который определяет до переданной точки расстояние, округлённое до сотых.

2. Создайте класс `PatchedPoint` — наследника класса `Point`. Требуется реализовать следующие виды инициализации нового класса:

параметров не передано — координаты точки равны 0;

передан один параметр — кортеж с координатами точки;

передано два параметра — координаты точки.

Реализуйте магические методы `_str_` и `_repr_`. При преобразовании в строку точка представляется в формате `(x, y)`.

Репрезентация же должна возвращать строку для инициализации точки двумя параметрами.

Реализуйте операторы `+` и `+=`. При выполнении кода `point + (x, y)`, создаётся новая точка, которая отличается от изначальной на заданное кортежем расстояние по осям `x` и `y`.

При выполнении кода `point += (x, y)` производится перемещение изначальной точки.

3. Запишите функцию `only_positive_even_sum`, которая принимает два параметра и возвращает их сумму. Если один из параметров не является целым числом, то следует вызвать исключение `TypeError`. Если один из параметров не является положительным чётным числом, то следует вызвать исключение `ValueError`.

Типовые задачи на использование библиотек для получения и обработки данных

1. Напишите функцию `stairs`, принимающую вектор и возвращающую матрицу, в которой каждая строка является копией вектора со смещением.

2. Напишите функцию `length_stats`, которая получает текст, а возвращает объект `Series` со словами в качестве индексов и их длинами в качестве значений. Все слова в тексте предварительно переведите в нижний регистр, избавьтесь от знаков препинания и цифр, а также отсортируйте в лексикографическом порядке.

3. Данные об успеваемости представлены `DataFrame` со столбцами:

`name` — имя;

`maths` — оценка по математике;

`physics` — оценка по физике;

`computer science` — оценка по информатике.

Напишите функцию `best`, которая фильтрует всех «ударников» в журнале.

Напишите функцию `need_to_work_better`, которая выбирает тех, у кого есть хотя бы одна двойка.

Напишите функцию `update`, которая добавляет к данным столбец `average`, содержащий среднюю оценку ученика, а также сортирует данные по убыванию этого столбца, а при равенстве средних — по имени лексикографически.

Критерии оценки

- «зачтено» повышенный уровень (81-100%)

Выполнены правильно все задания, представленные в описании лабораторных работ. Возможно наличие некоторых неточностей в решении задач, за что снижается балл за текущий контроль.

- «зачтено» пороговый уровень (60-80%)

Выполнено правильно более половины заданий, около третьей части заданий не выполнены. Возможно наличие некоторых неточностей в решении задач.

«не зачтено» уровень не сформирован (менее 60%)

Выполнено правильно менее половины заданий. Возможно наличие некоторых неточностей в решении задач.

5.3. Темы письменных работ (эссе, рефераты, курсовые работы и др.)

Не предусмотрено.

5.4. Оценочные средства для промежуточной аттестации

Примерный перечень вопросов к промежуточной аттестации 2 семестр (зачет с оценкой)

1. Типы данных.

2. Ветвления и циклы.

3. Векторы.

4. Строки.

5. Составные типы данных.

6. Ссылки, указатели, константность

7. Функции

8. Шаблоны

9. Файлы. Текстовые файлы. Файловые операции.

Примерный перечень вопросов к промежуточной аттестации 3 семестр (экзамен):

1. Указатели. Арифметика указателей.

2. Динамическое выделение памяти, динамические массивы.

3. Последовательные контейнеры

4. Ассоциативные контейнеры
5. Классы
6. Шаблонные классы
7. Наследование и полиморфизм
8. Модули.
9. Обработка исключений.
10. Идиома RAII и умные указатели

Примерный перечень вопросов к промежуточной аттестации 4 семестр (экзамен):

1. Общие сведения о языке Python и особенности его стиля программирования.
2. Синтаксис и управляющие конструкции языка Python.
3. Переменные, значения и их типы. Присваивание значения.
4. Ввод значений с клавиатуры.
5. Встроенные операции и функции. Основные алгоритмические конструкции.
6. Условный оператор. Множественное ветвление. Условия равенства / неравенства.
7. Циклы и счетчики.
8. Определение функций. Параметры и аргументы. Вызовы функций. Оператор возврата return.
9. Последовательности в Python. Операторы, общие для всех типов последовательностей.
10. Специальные операторы и функции для работы со списками.
11. Работа со словарями. Методы словарей.
12. Множества.
14. Списочные выражения. Модель памяти для типов языка Python
15. Рекурсия. Декораторы. Генераторы
16. Позиционные и именованные аргументы. Функции высших порядков. Лямбда-функции

Примерный перечень тем к промежуточной аттестации 5 семестр (экзамен):

1. Работа с базами данных. Основные понятия теории реляционных баз данных. Таблицы, поля, записи. Основные и внешние ключи.
2. Подключение к базе данных. ODBC источник данных. Понятие рекордсета.
3. Подготовка и выполнение SQL запросов к базе данных.

Примерный перечень вопросов и заданий к промежуточной аттестации 6 семестр (экзамен):

1. Объектная модель Python. Классы, поля и методы.
2. Волшебные методы, переопределение методов.
3. Наследование.
4. Модель исключений Python. Try, except, else, finally.
5. Модули.
6. Модуль numpy.
7. Модуль pandas.

Критерии оценки на экзамене/зачет с оценкой

- оценка «отлично» (повышенный уровень):

- 1) Студент показал прочные знания основных положений учебной дисциплины, умение самостоятельно решать конкретные практические задачи.
- 2) Подтверждает примерами теоретический материал.
- 3) Если полностью ответил на два вопроса.

- оценка «хорошо» (пороговый уровень):

Студент показал прочные знания основных положений учебной дисциплины, умение самостоятельно решать конкретные практические задачи. В ответе студент допускает неточности фактического и теоретического плана, однако может исправить их при уточнении преподавателем.

– оценка «удовлетворительно»:

Студент показал знание основных положений учебной дисциплины, умение получить с помощью преподавателя правильное решение конкретной практической задачи. В ответе на теоретические вопросы студент допускает ошибки, ответ неполный; в большинстве примеров практической части допускает ошибки, которые исправляет при помощи наводящих вопросов преподавателя.

- оценка «неудовлетворительно» (уровень не сформирован):

Студент не владеет теоретическими сведениями по указанным вопросам, затрудняется в приведении примеров, большая часть практического материала выполнена неверно, студент затрудняется в исправлении ошибок.

Контрольные тесты и задания:

Название вопроса: 1 (ОПК-6)

Формулировка вопроса: Как подключить стандартную библиотеку iostream?

Варианты ответов:

1. #include <iostream.h>

2. #include "iostream.h"

3. #include iostream

4. #include 'iostream.h'

Ключ: 1. #include <iostream.h>

Название вопроса: 2 (ОПК-6)

Формулировка вопроса: Как указать комментарий?

Варианты ответов:

1. // здесь комментарий

2. # здесь комментарий

3. / здесь комментарий

4. /* здесь комментарий */

Ключ: 1. // здесь комментарий.

Название вопроса: 3 (ОПК-6)

Формулировка вопроса: Где правильно указана переменная?

Варианты ответов:

1. done = true;

2. var str = "Hi";

3. char sym = 'a';

4. float x = 32,14;

Ключ: 3. char sym = 'a';

Название вопроса: 4 (ОПК-6)

Формулировка вопроса: Как выглядит функция ввода?

Варианты ответов:

1. int x; cin >> x;

2. int x; cin << x;

3. int x; cin x;

4. int x=input();

Ключ: 1. int x; cin >> x;

Название вопроса: 5 (ПК-2)

Формулировка вопроса: Как создать список в Python?

Варианты ответов:

1. list = {1, 2, 3}

2. list = [1, 2, 3]

3. list = (1, 2, 3)

4. list = <1, 2, 3>

Ключ: 2. list = [1, 2, 3]

Название вопроса: 6 (ПК-2)

Формулировка вопроса: Какой оператор используется для возведения числа в степень в Python?

Варианты ответов:

1. ^

2. *

3. **

4. //

Ключ: 3. **

Название вопроса: 7 (ПК-2)

Формулировка вопроса: Необходимо собрать и вывести все уникальные слова из строки рекламного текста. Какой из перечисленных типов данных Python подходит лучше всего?

Варианты ответов:

1. множество (set)

2. кортеж (tuple)

3. список (list)

4. словарь (dict)

Ключ: 1. множество (set)

Название вопроса: 8 (ПК-2)

Формулировка вопроса: Какой тип данных возвращает функция input() в Python?

Варианты ответов:

1. str

2. int

3. float

4. bool
Ключ: 1. str

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)				
6.1. Рекомендуемая литература				
6.1.1. Основная литература				
	Авторы, составители	Заглавие	Издательство, год	Эл. адрес
Л1.1	Лозовский, В. В., Морозов В. А., Гололобов А. А.	Объектно-ориентированное программирование: учебно-методическое пособие	Москва: РТУ МИРЭА, 2024	https://e.lanbook.com/book/405227
Л1.2	Кочетыгов А. А.	Основы программирования на языке Python: учебное пособие	Тула: ТулГУ, 2024	https://e.lanbook.com/book/427316
Л1.3	Сергеева О. А.	Программирование на Python: учебно- методическое пособие	Кемерово: КемГУ, 2024	https://e.lanbook.com/book/420758
Л1.4	Юрина Т. А.	Объектно-ориентированное программирование: учебно-методическое пособие	Омск: СибАДИ, 2023	https://e.lanbook.com/book/338576
Л1.5	Авсеева О. В., Чернышов М. К.	Объектно-ориентированное программирование. Часть 1: учебно- методическое пособие	Воронеж: ВГУ, 2021	https://e.lanbook.com/book/454691
6.1.2. Дополнительная литература				
	Авторы, составители	Заглавие	Издательство, год	Эл. адрес
Л2.1	Архипов О. Г., Батасова В. С., Гречкина [и др.] П. В., Маран М. М.	Программирование. Сборник задач: учебное пособие для вузов	Санкт-Петербург: Лань, 2025	https://e.lanbook.com/book/443291
Л2.2	Тонхоноева А. А.	Программирование на языке C++: учебно- методическое пособие	Улан-Удэ: БГУ, 2024	https://e.lanbook.com/book/432515
Л2.3	Ладанова Е. О.	Разработка предметно-ориентированных языков программирования: учебное пособие	Москва: РТУ МИРЭА, 2024	https://e.lanbook.com/book/448907
6.3.1 Перечень программного обеспечения				
6.3.1.1	MS Office			
6.3.1.2	MS WINDOWS			
6.3.1.3	Code::Blocks			
6.3.1.4	Dev-C++			
6.3.1.5	Visual Studio			
6.3.1.6	Python			
6.3.1.7	Python(x,y)			
6.3.1.8	Kaspersky Endpoint Security для бизнеса СТАНДАРТНЫЙ			
6.3.1.9	Moodle			
6.3.1.10	NVDA			
6.3.1.11	Яндекс.Браузер			
6.3.1.12	Far Manager			
6.3.1.13	Astra Linux			
6.3.1.14	LibreOffice			
6.3.1.15	StarUML			
6.3.1.16	РЕД ОС			
6.3.2 Перечень информационных справочных систем				
6.3.2.1	Электронно-библиотечная система «Издательство Лань»			
6.3.2.2	База данных «Электронная библиотека Горно-Алтайского государственного университета»			
6.3.2.3	Электронно-библиотечная система IPRbooks			

7. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

	лекция-визуализация	
	проблемная лекция	

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Номер аудитории	Назначение	Основное оснащение
209 Б1	Компьютерный класс. Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Помещение для самостоятельной работы	Рабочее место преподавателя. Посадочные места обучающихся (по количеству обучающихся). Маркерная ученическая доска, экран, мультимедиапроектор, компьютеры с доступом в Интернет
211 Б1	Компьютерный класс. Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Помещение для самостоятельной работы	Рабочее место преподавателя. Посадочные места обучающихся (по количеству обучающихся), компьютеры с доступом к Интернет
201 Б1	Кабинет методики преподавания информатики. Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Помещение для самостоятельной работы	Маркерная ученическая доска, экран, мультимедиапроектор. Рабочее место преподавателя. Посадочные места обучающихся (по количеству обучающихся), компьютеры с доступом к Интернет

9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)**Методические указания по освоению дисциплин (модулей)**

Лекции, с одной стороны – это одна из основных форм учебных занятий в высших учебных заведениях, представляющая собой систематическое, последовательное устное изложение преподавателем определенного раздела конкретной науки или учебной дисциплины, с другой – это особая форма самостоятельной работы с учебным материалом. Лекция не заменяет собой книгу, она только подталкивает к ней, раскрывая тему, проблему, выделяя главное, существенное, на что следует обратить внимание, указывает пути, которым нужно следовать, добиваясь глубокого понимания поставленной проблемы, а не общей картины.

Работа на лекции – это сложный процесс, который включает в себя такие элементы как слушание, осмысление и собственно конспектирование. Для того, чтобы лекция выполнила свое назначение, важно подготовиться к ней и ее записи еще до прихода преподавателя в аудиторию. Без этого дальнейшее восприятие лекции становится сложным. Лекция в университете рассчитана на подготовленную аудиторию. Преподаватель излагает любой вопрос, ориентируясь на те знания, которые должны быть у студентов, усвоивших материал всех предыдущих лекций. Важно научиться слушать преподавателя во время лекции, поддерживать непрерывное внимание к выступающему.

Однако, одного слушания недостаточно. Необходимо фиксировать, записывать тот поток информации, который сообщается во время лекции – научиться вести конспект лекции, где формулировались бы наиболее важные моменты, основные положения, излагаемые лектором. Для ведения конспекта лекции следует использовать тетрадь. Ведение конспекта на листочках не рекомендуется, поскольку они не так удобны в использовании и часто теряются. При оформлении конспекта лекции необходимо оставлять поля, где студент может записать свои собственные мысли, возникающие параллельно с мыслями, высказанными лектором, а также вопросы, которые могут возникнуть в процессе слушания, чтобы получить на них ответы при самостоятельной проработке материала лекции, при изучении рекомендованной литературы или непосредственно у преподавателя в конце лекции. Составляя конспект лекции, следует оставлять значительный интервал между строчками. Это связано с тем, что иногда возникает необходимость вписать в первоначальный текст лекции одну или несколько строчек, имеющих принципиальное значение и почерпнутых из других источников. Расстояние между строками необходимо также для подчеркивания слов или целых групп слов (такое подчеркивание вызывается необходимостью привлечь внимание к данному месту в тексте при повторном чтении). Обычно подчеркивают определения, выводы.

Также важно полностью без всяких изменений вносить в тетрадь схемы, таблицы, чертежи и т.п., если они предполагаются

в лекции. Для того, чтобы совместить механическую запись с почти дословным фиксированием наиболее важных положений, можно использовать системы условных сокращений. В первую очередь сокращаются длинные слова и те, что повторяются в речи лектора чаще всего. При этом само сокращение должно быть по возможности кратким.

Семинарские (практические) занятия Самостоятельная работа студентов по подготовке к семинарскому (практическому) занятию должна начинаться с ознакомления с планом семинарского (практического) занятия, который включает в себя вопросы, выносимые на обсуждение, рекомендации по подготовке к семинару (практическому занятию), рекомендуемую литературу к теме. Изучение материала следует начать с просмотра конспектов лекций. Восстановив в памяти материал, студент приводит в систему основные положения темы, вопросы темы, выделяя в ней главное и новое, на что обращалось внимание в лекции. Затем следует внимательно прочитать соответствующую главу учебника.

Для более углубленного изучения вопросов рекомендуется конспектирование основной и дополнительной литературы. Читая рекомендованную литературу, не стоит пассивно принимать к сведению все написанное, следует анализировать текст, думать над ним, этому способствуют записи по ходу чтения, которые превращают чтение в процесс. Записи могут вестись в различной форме: развернутых и простых планов, выписок (тезисов), аннотаций и конспектов.

Подобрав, отработав материал и усвоив его, студент должен начать непосредственную подготовку своего выступления на семинарском (практическом) занятии для чего следует продумать, как ответить на каждый вопрос темы.

По каждому вопросу плана занятий необходимо подготовиться к устному сообщению (5-10 мин.), быть готовым принять участие в обсуждении и дополнении докладов и сообщений (до 5 мин.).

Выступление на семинарском (практическом) занятии должно удовлетворять следующим требованиям: в нем излагаются теоретические подходы к рассматриваемому вопросу, дается анализ принципов, законов, понятий и категорий; теоретические положения подкрепляются фактами, примерами, выступление должно быть аргументированным.

Лабораторные работы являются основными видами учебных занятий, направленными на экспериментальное (практическое) подтверждение теоретических положений и формирование общепрофессиональных и профессиональных компетенций. Они составляют важную часть теоретической и профессиональной практической подготовки.

В процессе лабораторной работы как вида учебного занятия студенты выполняют одно или несколько заданий под руководством преподавателя в соответствии с изучаемым содержанием учебного материала.

При выполнении обучающимися лабораторных работ значимым компонентом становятся практические задания с использованием компьютерной техники, лабораторно - приборного оборудования и др. Выполнение студентами лабораторных работ проводится с целью: формирования умений, практического опыта (в соответствии с требованиями к результатам освоения дисциплины, и на основании перечня формируемых компетенций, установленными рабочей программой дисциплины), обобщения, систематизации, углубления, закрепления полученных теоретических знаний, совершенствования умений применять полученные знания на практике.

Состав заданий для лабораторной работы должен быть спланирован с расчетом, чтобы за отведенное время они могли быть выполнены качественно большинством студентов.

При планировании лабораторных работ следует учитывать, что в ходе выполнения заданий у студентов формируются умения и практический опыт работы с различными приборами, установками, лабораторным оборудованием, аппаратурой, программами и др., которые могут составлять часть профессиональной практической подготовки, а также исследовательские умения (наблюдать, сравнивать, анализировать, устанавливать зависимости, делать выводы и обобщения, самостоятельно вести исследование, оформлять результаты).

Выполнению лабораторных работ предшествует проверка знаний студентов - их теоретической готовности к выполнению задания.

Формы организации студентов при проведении лабораторных работ: фронтальная, групповая и индивидуальная. При фронтальной форме организации занятий все студенты выполняют одновременно одну и ту же работу. При групповой форме организации занятий одна и та же работа выполняется группами по 2 - 5 человек. При индивидуальной форме организации занятий каждый студент выполняет индивидуальное задание.

Текущий контроль учебных достижений по результатам выполнения лабораторных работ проводится в соответствии с системой оценивания (рейтинговой, накопительной и др.), а также формами и методами (как традиционными, так и инновационными, включая компьютерные технологии), указанными в рабочей программе дисциплины (модуля). Текущий контроль проводится в пределах учебного времени, отведенного рабочим учебным планом на освоение дисциплины, результаты заносятся в журнал учебных занятий.

Объем времени, отводимый на выполнение лабораторных работ, планируется в соответствии с учебным планом ОПОП.

Перечень лабораторных работ в РПД, а также количество часов на их проведение должны обеспечивать реализацию требований к знаниям, умениям и практическому опыту студента по дисциплине (модулю) соответствующей ОПОП.

Самостоятельная работа обучающихся – это планируемая учебная, учебно-исследовательская, научно-исследовательская работа, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Объем самостоятельной работы определяется учебным планом основной профессиональной образовательной программы (ОПОП), рабочей программой дисциплины (модуля).

Самостоятельная работа организуется и проводится с целью формирования компетенций, понимаемых как способность применять знания, умения и личностные качества для успешной практической деятельности, в том числе:

- формирования умений по поиску и использованию нормативной, правовой, справочной и специальной литературы, а также других источников информации;
- качественного освоения и систематизации полученных теоретических знаний, их углубления и расширения по применению на уровне межпредметных связей;
- формирования умения применять полученные знания на практике (в профессиональной деятельности) и закрепления практических умений обучающихся;
- развития познавательных способностей, формирования самостоятельности мышления обучающихся;
- совершенствования речевых способностей обучающихся;
- формирования необходимого уровня мотивации обучающихся к систематической работе для получения знаний, умений и

владений в период учебного семестра, активности обучающихся, творческой инициативы, самостоятельности, ответственности и организованности;

- формирования способностей к саморазвитию (самопознанию, самоопределению, самообразованию, самосовершенствованию, самореализации и саморегуляции);
- развития научно-исследовательских навыков;
- развития навыков межличностных отношений.

К самостоятельной работе по дисциплине (модулю) относятся: проработка теоретического материала дисциплины (модуля); подготовка к семинарским и практическим занятиям, в т.ч. подготовка к текущему контролю успеваемости обучающихся (текущая аттестация); подготовка к лабораторным работам; подготовка к промежуточной аттестации (зачётам, экзаменам).

Виды, формы и объёмы самостоятельной работы обучающихся при изучении дисциплины (модуля) определяются:

- содержанием компетенций, формируемых дисциплиной (модулем);
- спецификой дисциплины (модуля), применяемыми образовательными технологиями;
- трудоёмкостью СР, предусмотренной учебным планом;
- уровнем высшего образования (бакалавриат, специалитет, магистратура, аспирантура), на котором реализуется ОПОП;
- степенью подготовленности обучающихся.

Курсовая работа является самостоятельным творческим письменным научным видом деятельности студента по разработке конкретной темы. Она отражает приобретенные студентом теоретические знания и практические навыки. Курсовая работа выполняется студентом самостоятельно под руководством преподавателя.

Курсовая работа, наряду с экзаменами и зачётами, является одной из форм контроля (аттестации), позволяющей определить степень подготовленности будущего специалиста. Курсовые работы защищаются студентами по окончании изучения указанных дисциплин, определенных учебным планом.

Оформление работы должно соответствовать требованиям. Объём курсовой работы: 25–30 страниц. Список литературы и Приложения в объём работы не входят. Курсовая работа должна содержать: титульный лист, содержание, введение, основную часть, заключение, список литературы, приложение (при необходимости). Курсовая работа подлежит рецензированию руководителем курсовой работы. Рецензия является официальным документом и прикладывается к курсовой работе.

Тематика курсовых работ разрабатывается в соответствии с учебным планом. Руководитель курсовой работы лишь помогает студенту определить основные направления работы, очертить её контуры, указывает те источники, на которые следует обратить главное внимание, разъясняет, где отыскать необходимые книги.

Составленный список источников научной информации, подлежащий изучению, следует показать руководителю курсовой работы.

Курсовая работа состоит из глав и параграфов. Вне зависимости от решаемых задач и выбранных подходов структура работы должна содержать: титульный лист, содержание, введение, основную часть; заключение; список литературы; приложение(я).

Во введении необходимо отразить: актуальность; объект; предмет; цель; задачи; методы исследования; структура работы.

Основную часть работы рекомендуется разделить на 2 главы, каждая из которых должна включать от двух до четырех параграфов.

Содержание глав и их структура зависит от темы и анализируемого материала.

Первая глава должна иметь обзорно–аналитический характер и, как правило, является теоретической.

Вторая глава по большей части раскрывает насколько это возможно предмет исследования. В ней приводятся практические данные по проблематике темы исследования.

Выводы оформляются в виде некоторого количества пронумерованных абзацев, что придает необходимую стройность изложению изученного материала. В них подводится итог проведённой работы, непосредственно выводы, вытекающие из всей работы и соответствующие выявленным проблемам, поставленным во введении задачам работы; указывается, с какими трудностями пришлось столкнуться в ходе исследования.

Правила написания и оформления курсовой работы регламентируются Положением о курсовой работе (проекте), утвержденным решением Ученого совета ФГБОУ ВО ГАГУ от 27 апреля 2017 г.