

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Астраханский государственный университет имени В. Н. Татищева»
(Астраханский государственный университет им. В. Н. Татищева)

СОГЛАСОВАНО
Руководитель ОПОП
_____ И.М.
Ажмухамедов

«02» июня 2023 г.

УТВЕРЖДАЮ
Заведующий кафедрой ЦТ

_____ А.Н. Марьенков

«02» июня 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Основы программирования

наименование

Составитель(-и)	Марьенков А.Н., к.т.н., доцент, заведующий кафедрой цифровых технологий
Направление подготовки	10.03.01 Информационная безопасность
Направленность (профиль) ОПОП	«Организация и технологии защиты информации (в сфере информационных и коммуникационных технологий)»
Квалификация (степень)	бакалавр
Форма обучения	очная
Год приема	2023
Курс	1, 2
Семестр	2, 3

Астрахань, 2023 г.

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

1.1. Целью освоения дисциплины (модуля) «Основы программирования» – является освоение базовых понятий, методов и приемов программирования, применяемых на всех основных этапах жизненного цикла программы, а также подготовка студентов к деятельности, связанной с использованием современных информационных технологий и с разработкой программного обеспечения для решения профессиональных задач.

1.2. Задачи освоения дисциплины (модуля):

- создание у студентов упорядоченной системы знаний о реальных возможностях программирования на языках высокого уровня, о направлениях профессионального программирования, тенденциях развития объектной технологии построения программных приложений;
- формирование базы знаний для работы в интегрированных системах программирования;
- ознакомление студентов с практикой применения инструментария интегрированных сред программирования для решения различных прикладных задач

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОПОП

2.1. Учебная дисциплина (модуль) Б1.Б.08.01 «Основы программирования» входит в обязательную часть учебного плана 2023 года приема, рассчитана на два семестра (2, 3 семестры) и предусматривает сдачу студентами экзамена после первого и второго семестра на основе балльно-рейтинговой системы оценивания.

Общая трудоемкость дисциплины – 8 кредита (ЗЕ).

2.2. Для изучения данной учебной дисциплины (модуля) необходимы следующие знания, умения, навыки и (или) опыт деятельности, формируемые предшествующими дисциплинами:

знания:

- основные принципы алгоритмизации; основные методы обработки данных;
- этапы разработки программ и методы автоматизации программирования;
- основные понятия и методы технологии программирования;
- конструкции языка высокого уровня;

умения:

- самостоятельно работать на ПЭВМ с соблюдением основных принципов работы;
 - осуществлять декомпозицию решения задачи и составлять алгоритмы отдельных его частей в соответствии с современной технологией программирования;
 - применять основные операторы, общие для всех языков программирования;
 - использовать отладчик как средство изучения и тестирования программ;
- работать с ресурсами компьютера программными средствами;

навыки:

- навыками разработки и отладки программ на языках высокого уровня, навыками оптимизации программного кода.

2.3. Последующие учебные дисциплины (модули) и (или) практики, для которых необходимы знания, умения, навыки, формируемые данной учебной дисциплиной (модулем):

- Технологии и методы программирования.
- Дискретная математика.
- Теория информации.
- Вероятностно-статистические методы в анализе данных.

- Методы и средства криптографической защиты информации.
- Теория принятия решений и методы оптимизации.
- Математическая логика и теория алгоритмов.

Также дисциплина «Основы программирования» поможет студентам при написании бакалаврской работы.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Процесс освоения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВО и ОПОП ВО по данному направлению подготовки (специальности):

а) общепрофессиональных (ОПК): ОПК 7 – способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности.

Таблица 1 –Декомпозиция результатов обучения

Код и наименование компетенции	Планируемые результаты обучения по дисциплине (модулю)		
	Знать (1)	Уметь (2)	Владеть (3)
ОПК 7 – способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности	ОПК-7.1. Знать: основы программирования.	ОПК-7.2. Уметь: использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности.	ОПК-7.3. Владеть: навыками программирования для решения задач профессиональной деятельности.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Общая трудоемкость дисциплины – 4 ЗЕ во 2-м семестре и 4 ЗЕ в 3-семестре. На освоение дисциплины отводится 288 часов, 90 часов выделено на контактную работу обучающихся с преподавателем (лекции – 36, лабораторные работы –54), 198 часов – на самостоятельную работу обучающихся.

Таблица 2– Структура и содержание дисциплины (модуля)

№ п/п	Наименование раздела (темы)	Семестр	Неделя семестра	Контактная работа (в часах)			Самостоят. работа		Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				Л	ПЗ	ЛР	КР	СР	
1.	Нахождение значения выражения на языке программирования Python.	2	1-2	2		2		12	Отчет по лабораторной работе 1
2.	Линейные алгоритмы. Базовые конструкции		3-4	2		2		12	Отчет по лабораторной

№ п/п	Наименование раздела (темы)	Семестр	Неделя семестра	Контактная работа (в часах)			Самостоят. работа		Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточн ой аттестации (по семестрам)
				Л	ПЗ	ЛР	КР	СР	работе 1
	языка Python.								работе 1
3.	Операции, выражения в языке Python. Решение задач с помощью условных операторов.		5-6	2		2		12	Отчет по лабораторной работе 2,
4.	Решение задач с помощью циклов.		7-8	2		2		12	Отчет по лабораторной работе 3, ,
5.	Одномерные массивы. Двумерные массивы.		9-10	2		2		12	Отчет по лабораторной работе 4
6.	Строковые данные.		11-12	2		2		12	Отчет по лабораторной работе 5, Контрольная работа 1 Проект
7.	Функции.		13-14	2		2		12	Отчет по лабораторной работе 6, Отчет по лабораторной работе 7
8.	Рекурсивные функции.		15-16	2		2		12	Отчет по лабораторной работе 8 Проект
9.	Работа с файлами.		17-18	2		2		12	Отчет по лабораторной работе 9. Отчет по лабораторной работе 10. Отчет по

№ п/п	Наименование радела (темы)	Семестр	Неделя семестра	Контактная работа (в часах)			Самостоят. работа		Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточн ой аттестации (по семестрам)
				Л	ПЗ	ЛР	КР	СР	
									лабораторной работе 11. Контрольная работа 2 Проект
	Итог за 2 семестр			18		18		108	Экзамен
10.	Основы объектно-ориентированного программирования.	3	1-2	2		4		10	Отчет по лабораторной работе 12
11.	Основные понятия объектно-ориентированного программирования (ООП).		3-4	2		4		10	Отчет по лабораторной работе 12
12.	Классы.		5-6	2		4		10	Отчет по лабораторной работе 13, Контрольная работа 3
13.	Поля и методы класса.		7-8	2		4		10	Отчет по лабораторной работе 14
14.	Перегрузка операторов.		9-10	2		4		10	Отчет по лабораторной работе 15
15.	Наследование.		11-12	2		4		10	Отчет по лабораторной работе 16, Проект
16.	Виртуальные функции. Абстрактные классы.		13-14	2		4		10	Отчет по лабораторной работе 17, Контрольная работа 4
17.	Исключительные ситуации. Классы исключений.		15-16	2		4		10	Отчет по лабораторной работе 18, Проект
18.	Шаблоны классов.		17-18	2		4		10	Отчет по лабораторной работе 19, Контрольная

№ п/п	Наименование радела (темы)	Семестр	Неделя семестра	Контактная работа (в часах)			Самостоят. работа		Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточн ой аттестации (по семестрам)
				Л	ПЗ	ЛР	КР	СР	
									работа 5
	Итого за 3 семестр			18		36		90	Проект Экзамен
	Итого			36		54		198	

Примечание: Л – лекция; ПЗ – практическое занятие, семинар; ЛР – лабораторная работа; КР – курсовая работа; СР – самостоятельная работа.

Таблица 3– Матрица соотнесения тем/разделов учебной дисциплины/модуля и формируемых компетенций

Темы, разделы дисциплины	Кол-во часов	Код компетенц ии	Σ общее количество компетенци й
		ОПК 7	
Нахождение значения выражения на языке программирования Python.	16	+	1
Линейные алгоритмы. Базовые конструкции языка Python.	16	+	1
Операции, выражения в языке Python. Решение задач с помощью условных операторов.	16	+	1
Решение задач с помощью циклов.	16	+	1
Одномерные массивы. Двумерные массивы.	16	+	1
Строковые данные.	16	+	1
Функции.	16	+	1
Рекурсивные функции.	16	+	1
Работа с файлами.	16	+	1
Основы объектно-ориентированного программирования.	16	+	1

Основные понятия объектно-ориентированного программирования (ООП).	16	+	1
Классы.	16	+	1
Поля и методы класса.	16	+	1
Перегрузка операторов.	16	+	1
Наследование.	16	+	1
Виртуальные функции. Абстрактные классы.	16	+	1
Исключительные ситуации. Классы исключений.	16	+	1
Шаблоны классов.	16	+	1
ИТОГО	288		1

Содержание дисциплины

Тема 1. Нахождение значения выражения на языке программирования Python. Построение блок-схем, проектирование блок-схем для различных задач. Изучение среды разработки на языке программирования Python.

Тема 2. Линейные алгоритмы. Базовые конструкции языка Python. Алфавит, идентификаторы, служебные слова, константы. Системы счисления. Целочисленные и вещественные типы данных, представление целых и вещественных чисел в компьютере. Написание простейших программ.

Тема 3. Операции, выражения в языке Python. Решение задач с помощью условных операторов. Операторы последовательного выполнения, операторы выбора. Основы структурного программирования.

Тема 4. Решение задач с помощью циклов. Операторы цикла, операторы передачи управления. Продолжение изучения основ структурного программирования. Вычисления выражений с использованием циклов.

Тема 5. Одномерные массивы. Двумерные массивы. Изучить особенности работы с типом данных массив на языке Python.

Тема 6. Строковые данные. Изучение особенности работы со строками на языке Python. Написание программ, обрабатывающих строковые данные.

Тема 7. Функции. Определение, описание и вызов функции. Способы передачи параметров в функцию. Параметры по умолчанию. Параметры функции main. Функции с переменным числом параметров.

Тема 8. Рекурсивные функции. Изучение особенностей использования рекурсивных функций в языке программирования Python.

Тема 9. Работа с файлами. Изучение особенности работы с файлами в программах на Python. Последовательный и произвольный доступ. Создание, редактирование, чтение, копирование и удаление файлов. Поиск в файлах.

Тема 10. Основы объектно-ориентированного программирования. Понятия абстракции, абстрактного типа данных.

Тема 11. Основные понятия объектно-ориентированного программирования (ООП). Инкапсуляция, наследование и полиморфизм.

Тема 12. Классы. Понятие класса. Доступ к компонентам класса. Конструкторы и деструкторы.

Тема 13. Поля и методы класса. Указатель this. Статические компоненты класса.

Дружественные функции и классы.

Тема 14. Перегрузка операторов. Наследование. Полиморфизм.

Тема 15. Виртуальные функции. Абстрактные классы.

Тема 16. Поточковые средства ввода/вывода данных. Иерархия классов потоков в языке C++. Форматированный ввод/вывод. Файловый ввод/вывод.

Тема 17. Исключительные ситуации. Основные понятия. Механизм обработки исключений. Классы исключений.

Тема 18. Шаблоны классов. Определение шаблона классов, типы параметров шаблона. Инстанцирование шаблона. Специализация шаблона.

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРЕПОДАВАНИЮ И ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1. Указания для преподавателей по организации и проведению учебных занятий по дисциплине (модулю):

При подготовке к лекционным занятиям необходимо воспользоваться учебно-методической литературой. Лекции необходимо проводить с использованием презентаций, созданных в Microsoft PowerPoint.

При подготовке к лабораторным занятиям необходимо воспользоваться учебно-методической литературой, а также пользоваться ресурсами сети Интернет.

Методические рекомендации по выполнению лабораторных и контрольных работ, проведению экзамена

Отчет по лабораторной работе

Отчет по лабораторной работе представляется в электронном виде. Защита отчета проходит в форме доклада студента по выполненной работе и ответов на вопросы преподавателя. В случае, если оформление отчета и поведение студента во время защиты соответствуют указанным требованиям, студент получает максимальное количество баллов.

Основаниями для снижения количества баллов в диапазоне от max до min являются:

- отсутствие списка использованной литературы,
- небрежное выполнение,
- отсутствие выводов.

Отчет не может быть принят и подлежит доработке в случае:

- отсутствия необходимых разделов,
- отсутствия необходимого графического материала,
- неверных результатов расчета.

В отчете по выполненной лабораторной работе должны быть указаны:

- тема лабораторной работы,
- пакет документов в соответствии с темой лабораторной работы,
- использованная литература.

Экзамен

Экзамен заключается в письменном ответе на 2 теоретических вопроса и устном собеседовании по каждому теоретическому вопросу.

Основаниями для снижения оценки за теоретический вопрос являются:

- небрежное выполнение;
- неполный ответ;
- наличие мелких неточностей или незначительных искажений фактов;
- неточные объяснения при собеседовании;
- отсутствие ответов на заданные при собеседовании вопросы.

Оценивание студентов на экзамене осуществляется в соответствии с требованиями и

критериями 100-балльной шкалы. Учитываются как результаты текущего контроля, так и знания, навыки и умения, непосредственно показанные студентами в ходе экзамена.

5.2. Указания для обучающихся по освоению дисциплины (модулю)

Во время самостоятельной работы необходимо воспользоваться учебно-методической литературой из п.8 № 1– 30.

Таблица 4 – Содержание самостоятельной работы обучающихся

<i>Номер раздела (темы)</i>	<i>Темы/вопросы, выносимые на самостоятельное изучение</i>	<i>Кол-во часов</i>	<i>Формы работы</i>
Тема 1.	Язык программирования Python Характеристика языка, стандарты.	12	Внеаудиторная, изучение учебных пособий
Тема 2.	Базовые конструкции языка Python	12	Внеаудиторная, изучение учебных пособий
Тема 3.	Операции, выражения в языке Python. Операторы последовательного выполнения, операторы выбора, операторы цикла, операторы передачи управления. Указатели и ссылки.	12	Внеаудиторная, изучение учебных пособий
Тема 4.	Массивы. Строки.	12	Внеаудиторная, изучение учебных пособий
Тема 5.	Структуры, объединения, битовые поля.	12	Внеаудиторная, изучение учебных пособий
Тема 6.	Функции.	12	Внеаудиторная, изучение учебных пособий
Тема 7.	Потоковые средства ввода/вывода данных.	12	Внеаудиторная, изучение учебных пособий
Тема 8.	Классы памяти.	12	Внеаудиторная, изучение учебных пособий
Тема 9.	Условная компиляция. Макроподстановки. Операции с разрядами. Поразрядные логические операции и операции сдвига.	12	Внеаудиторная, изучение учебных пособий
Тема 10.	Основы объектно-ориентированного программирования.	10	Внеаудиторная, изучение учебных пособий
Тема 11.	Основные понятия объектно-ориентированного программирования (ООП).	10	Внеаудиторная, изучение учебных пособий
Тема 12.	Классы.	10	Внеаудиторная, изучение учебных пособий
Тема 13.	Поля и методы класса.	10	Внеаудиторная, изучение учебных пособий

			пособий
Тема 14.	Перегрузка операторов.	10	Внеаудиторная, изучение учебных пособий
Тема 15.	Наследование.	10	Внеаудиторная, изучение учебных пособий
Тема 16.	Виртуальные функции. Абстрактные классы.	10	Внеаудиторная, изучение учебных пособий
Тема 17.	Исключительные ситуации. Классы исключений.	10	Внеаудиторная, изучение учебных пособий
Тема 18.	Шаблоны классов.	10	Внеаудиторная, изучение учебных пособий

5.3. Виды и формы письменных работ, предусмотренных при освоении дисциплины, выполняемые обучающимися самостоятельно - проект.

Задание для выполнения проекта

Студенты в составе группы реализуют проект по разработке программного обеспечения. В задачи проекта входят: поиск заказчика и формулировка проекта, генерация требований к проекту, проектирование, разработка, тестирование и документирование программного обеспечения; презентация результатов проекта

Студенты разбиваются на группы по 4-6 человек и реализуют проект по разработке программного обеспечения. На первом этапе группа должны найти заказчика проекта, определить подлежащую решению с помощью разрабатываемого ПО проблему и сформулировать тематику проекта. Заказчиком может являться произвольная бизнес-структура либо одно из подразделений университета. Проект должен быть направлен на решение реальной задачи.

Работа над проектом предполагает использование гибких методологий разработки ПО, но при этом не исключает полностью этапа предварительного эскизного проектирования.

При работе над проектом студенты осуществляют сбор и документирование требований к проекту, строят базовые модели системы с использованием UML.

При разработке проекта студенты должны использовать методы обеспечения качества программного продукта: модульное и интеграционное тестирование, системы управления версиями, системы отслеживания ошибок.

По завершении проекта на него подготавливается программная документация и студенты выполняют презентацию результатов проекта.

Критерии оценки проекта:

– оценка «отлично» выставляется обучающемуся, если студент представил проект в соответствии с методическими указаниями, информация в проекте сформулирована обоснованно, логично и последовательно, применен творческий подход, учтены основные нормативно-правовые документы по информационной безопасности;

– оценка «хорошо» выставляется обучающемуся, если студент представил проект в соответствии с методическими указаниями, информация в проекте сформулирована обоснованно, формулировки конкретные, приведены ссылки на нормативно-правовые

документы по информационной безопасности, допущены некоторые неточности, имеется одна негрубая ошибка.

– оценка «удовлетворительно» выставляется обучающемуся, если студент представил проект в соответствии с методическими указаниями, информация в проекте сформулирована с нарушением логики, не полная, формулировка общая или неполная, имеются одна или две негрубые ошибки, приведены неверные ссылки на нормативно-правовые документы по информационной безопасности;

– оценка «неудовлетворительно» выставляется обучающемуся, если студент не представил проект или выполнил ее неверно, без использования методических указаний, обоснования неверные, сделаны грубые ошибки, отсутствуют ссылки на нормативно-правовые документы по информационной безопасности.

6. ОБРАЗОВАТЕЛЬНЫЕ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

При реализации различных видов учебной работы по дисциплине могут использоваться электронное обучение и дистанционные образовательные технологии.

6.1. Образовательные технологии

Таблица 5 – Образовательные технологии, используемые при реализации учебных занятий

Раздел, тема дисциплины (модуля)	Форма учебного занятия		
	Лекция	Практическое занятие, семинар	Лабораторная работа
Нахождение значения выражения на языке программирования Python.	<i>Обзорная лекция</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>
Линейные алгоритмы. Базовые конструкции языка Python.	<i>Лекция-диалог</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>
Операции, выражения в языке Python. Решение задач с помощью условных операторов.	<i>Лекция</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>
Решение задач с помощью циклов.	<i>Обзорная лекция</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>
Одномерные массивы. Двумерные массивы.	<i>Лекция</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>
Строковые данные.	<i>Лекция-диалог</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>
Функции.	<i>Обзорная лекция</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>
Рекурсивные функции.	<i>Лекция-диалог</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>
Работа с файлами.	<i>Лекция</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>

Основы объектно-ориентированного программирования.	<i>Обзорная лекция</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>
Основные понятия объектно-ориентированного программирования (ООП).	<i>Лекция</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>
Классы.	<i>Лекция-диалог</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>
Поля и методы класса.	<i>Обзорная лекция</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>
Перегрузка операторов.	<i>Лекция</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>
Наследование.	<i>Лекция-диалог</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>
Виртуальные функции. Абстрактные классы.	<i>Лекция</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>
Исключительные ситуации. Классы исключений.	<i>Лекция-диалог</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>
Шаблоны классов.	<i>Лекция</i>	<i>Не предусмотрено</i>	<i>выполнение лабораторной работы</i>

Цели курса достигаются путём сочетания комплекса методов обучения, включающих лекции и лабораторные работы, выполняемые на ЭВМ.

Лабораторные работы на ЭВМ ориентированы на формирование деятельностных компетентностей. Они заключаются в выполнении цикла лабораторных работ. В процессе выполнения лабораторных работ достигаются следующие цели:

- закрепляются теоретические познания, полученные на лекциях, актуализируется их практическая значимость, закрепляется мотивация к освоению курса;
- студент приобретает практические навыки программирования на языках высокого уровня;
- приобретаются начальные навыки использования сред разработки программных проектов;
- формируется навык выявления ошибочных и нештатных ситуаций и реагирования на них.

Учебные занятия по дисциплине могут проводиться с применением информационно-телекоммуникационных сетей при опосредованном (на расстоянии) интерактивном взаимодействии обучающихся и преподавателя в режимах on-line в формах: видеолекций, лекций-презентаций, видеоконференции, собеседования в режиме чат, форума, чата, выполнения виртуальных практических и/или лабораторных работ и др.

Максимальный объем занятий обучающегося с применением электронных образовательных технологий не должен превышать 25%.

6.2. Информационные технологии

При реализации различных видов учебной и внеучебной работы используются следующие информационные технологии:

- использование возможностей Интернета в учебном процессе (использование информационного сайта преподавателя (рассылка заданий, предоставление выполненных работ, ответы на вопросы, ознакомление учащихся с оценками и т.д.));
- использование электронных учебников и различных сайтов (например, электронные библиотеки, журналы и т.д.) как источников информации;
- использование возможностей электронной почты преподавателя;
- использование средств представления учебной информации (электронных учебных пособий и практикумов, применение новых технологий для проведения очных (традиционных) лекций и семинаров с использованием презентаций и т.д.);
- использование интегрированных образовательных сред, где главной составляющей являются не только применяемые технологии, но и содержательная часть, т.е. информационные ресурсы (доступ к мировым информационным ресурсам, на базе которых строится учебный процесс);
- использование виртуальной обучающей среды (или системы управления обучением LMS Moodle «Электронное образование») или иных информационных систем, сервисов и мессенджеров]

6.3. Программное обеспечение, современные профессиональные базы данных и информационные справочные системы

6.3.1. Программное обеспечение

Наименование программного обеспечения	Назначение
Adobe Reader	Программа для просмотра электронных документов
Платформа дистанционного обучения LMS Moodle	Виртуальная обучающая среда
Mozilla FireFox	Браузер
Microsoft Office 2013, Microsoft Office Project 2013 , Microsoft Office Visio 2013	Офисная программа
7-zip	Архиватор
Microsoft Windows 7 Professional	Операционная система
Kaspersky Endpoint Security	Средство антивирусной защиты
Microsoft Visual C++ Express Edition, Microsoft Visual C# Express Edition	Среда разработки ПО

6.3.2. Современные профессиональные базы данных и информационные справочные системы

1. Электронный каталог Научной библиотеки АГУ на базе MARK SQL НПО «Информ-систем»: <https://library.asu.edu.ru>.
2. Электронный каталог «Научные журналы АГУ»: <http://journal.asu.edu.ru/>.
3. Универсальная справочно-информационная полнотекстовая база данных периодических изданий ООО «ИВИС»: <http://dlib.eastview.com/>
4. Электронно-библиотечная система elibrary. <http://elibrary.ru>
5. Справочная правовая система КонсультантПлюс: <http://www.consultant.ru>
6. Информационно-правовое обеспечение «Система ГАРАНТ»: <http://garant-astrakhan.ru>

7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

7.1. Паспорт фонда оценочных средств

При проведении текущего контроля и промежуточной аттестации по дисциплине (модулю) «Основы программирования» проверяется сформированность у обучающихся компетенций, указанных в разделе 3 настоящей программы. Этапность формирования данных компетенций в процессе освоения образовательной программы определяется последовательным освоением дисциплин (модулей) и прохождением практик, а в процессе освоения дисциплины (модуля) – последовательным достижением результатов освоения содержательно связанных между собой разделов, тем.

Таблица 6 – Соответствие разделов, тем дисциплины (модуля), результатов обучения по дисциплине (модулю) и оценочных средств

№ п/п	Контролируемые разделы дисциплины (модуля)	Код контролируемой компетенции (компетенций)	Наименование оценочного средства
1	Нахождение значения выражения на языке программирования Python.	ОПК 7	Отчет по лабораторной работе 1
2	Линейные алгоритмы. Базовые конструкции языка Python.	ОПК 7	Отчет по лабораторной работе 1
3	Операции, выражения в языке Python. Решение задач с помощью условных операторов.	ОПК 7	Отчет по лабораторной работе 2,
4	Решение задач с помощью циклов.	ОПК 7	Отчет по лабораторной работе 3, ,
5	Одномерные массивы. Двумерные массивы.	ОПК 7	Отчет по лабораторной работе 4
6	Строковые данные.	ОПК 7	Отчет по лабораторной работе 5, Контрольная работа 1 Проект

7	Функции.	ОПК 7	Отчет по лабораторной работе 6, Отчет по лабораторной работе 7
8	Рекурсивные функции.	ОПК 7	Отчет по лабораторной работе 8 Проект
9	Работа с файлами.	ОПК 7	Отчет по лабораторной работе 9. Отчет по лабораторной работе 10. Отчет по лабораторной работе 11. Контрольная работа 2 Проект
10	Основы объектно-ориентированного программирования.	ОПК 7	Отчет по лабораторной работе 12
11	Основные понятия объектно-ориентированного программирования (ООП).	ОПК 7	Отчет по лабораторной работе 12
12	Классы.	ОПК 7	Отчет по лабораторной работе 13, Контрольная работа 3
13	Поля и методы класса.	ОПК 7	Отчет по лабораторной работе 14
14	Перегрузка операторов.	ОПК 7	Отчет по лабораторной работе 15
15	Наследование.	ОПК 7	Отчет по лабораторной работе 16, Проект
16	Виртуальные функции. Абстрактные классы.	ОПК 7	Отчет по лабораторной работе 17, Контрольная работа 4
17	Исключительные ситуации. Классы исключений.	ОПК 7	Отчет по лабораторной работе 18, Проект
18	Шаблоны классов.	ОПК 7	Отчет по лабораторной работе 19, Контрольная работа 5 Проект

Рекомендуемые типы контроля для оценивания результатов обучения.

Для оценивания результатов обучения в виде **знаний** используются следующие типы контроля:

- тестирование;
- индивидуальное собеседование,
- письменные ответы на вопросы.

Тестовые задания должны охватывать содержание всего пройденного материала. Индивидуальное собеседование, письменная работа проводятся по разработанным вопросам

по отдельному учебному элементу программы (дисциплине).

Для оценивания результатов обучения в виде **умений и владений** используются следующие типы контроля:

- практические контрольные задания (далее – ПКЗ), включающих одну или несколько задач (вопросов) в виде краткой формулировки действий (комплекса действий), которые следует выполнить, или описание результата, который нужно получить.

По сложности ПКЗ разделяются на простые и комплексные задания.

Простые ПКЗ предполагают решение в одно или два действия. К ним можно отнести: простые ситуационные задачи с коротким ответом или простым действием; несложные задания по выполнению конкретных действий. Простые задания применяются для оценки умений. Комплексные задания требуют многоходовых решений как в типичной, так и в нестандартной ситуациях. Это задания в открытой форме, требующие поэтапного решения и развернутого ответа, в т.ч. задания на индивидуальное или коллективное выполнение проектов, на выполнение практических действий или лабораторных работ. Комплексные практические задания применяются для оценки владений.

Типы практических контрольных заданий:

- задания на установление правильной последовательности, взаимосвязанности действий, выяснения влияния различных факторов на результаты выполнения задания;
- установление последовательности (описать алгоритм выполнения действия),
- нахождение ошибок в последовательности (определить правильный вариант последовательности действий);
- указать возможное влияние факторов на последствия реализации умения и т.д.
- задания на принятие решения в нестандартной ситуации (ситуации выбора, многоальтернативности решений, проблемной ситуации);
- задания на оценку последствий принятых решений;
- задания на оценку эффективности выполнения действия.

7.2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

При решении комплексной ситуационной задачи можно использовать следующие критерии оценки:

Таблица 7 – Показатели оценивания результатов обучения в виде знаний

Шкала оценивания	Критерии оценивания
5 «отлично»	демонстрирует глубокое знание теоретического материала, умение обоснованно излагать свои мысли по обсуждаемым вопросам, способность полно, правильно и аргументированно отвечать на вопросы, приводить примеры
4 «хорошо»	демонстрирует знание теоретического материала, его последовательное изложение, способность приводить примеры, допускает единичные ошибки, исправляемые после замечания преподавателя
3 «удовлетворительно»	демонстрирует неполное, фрагментарное знание теоретического материала, требующее наводящих вопросов преподавателя, допускает существенные ошибки в его изложении, затрудняется в приведении примеров и формулировке выводов
2 «неудовлетворительно»	демонстрирует существенные пробелы в знании теоретического материала, не способен его изложить и ответить на наводящие вопросы преподавателя, не может привести примеры

Таблица 8 – Показатели оценивания результатов обучения в виде умений и владений

Шкала оценивания	Критерии оценивания
5 «отлично»	демонстрирует способность применять знание теоретического материала при выполнении заданий, последовательно и правильно выполняет задания, умеет обоснованно излагать свои мысли и делать необходимые выводы
4 «хорошо»	демонстрирует способность применять знание теоретического материала при выполнении заданий, последовательно и правильно выполняет задания, умеет обоснованно излагать свои мысли и делать необходимые выводы, допускает единичные ошибки, исправляемые после замечания преподавателя
3 «удовлетворительно»	демонстрирует отдельные, несистематизированные навыки, не способен применить знание теоретического материала при выполнении заданий, испытывает затруднения и допускает ошибки при выполнении заданий, выполняет задание при подсказке преподавателя, затрудняется в формулировке выводов
2 «неудовлетворительно»	не способен правильно выполнить задание

7.3. Контрольные задания или иные материалы, необходимые для оценки результатов обучения по дисциплине (модулю)

Тема 1. Нахождение значения выражения на языке программирования Python.

Линейные алгоритмы. Базовые конструкции языка Python.

Лабораторно-практическая работа 1. Базовые конструкции языка Python.

Вариант 1

$$\frac{x^3 - 7x + 10}{x^2 - 8x + 12}$$

$$\frac{2 - \cos 2x + \sin 2x}{1 + \cos 2x + \sin 2x}$$

3) Заданы координаты трех вершин треугольника (x_1, y_1) , (x_2, y_2) , (x_3, y_3) . Найти его периметр и площадь.

Вариант 2

$$\frac{5x - 10y}{2y} : \frac{6y - 3x}{8y^2}$$

$$\frac{2\sin(a+b) - \cos a \cdot \sin b}{\cos(a-b) - \sin a \cdot \sin b}$$

3) Дана сторона равностороннего треугольника. Найти площадь этого треугольника, его высоты, радиусы вписанной и описанной окружностей.

Вариант 3

$$1) \frac{5b - 10}{4 - 4b - b^2};$$

$$2) \ln(x^3 + 4\sqrt{x} + 5)$$

Найти сумму членов арифметической прогрессии, если известны ее первый член, знаменатель и число членов прогрессии.

Вариант 4

$$1) xy - \frac{x^3 - y^3}{x^2 + 2xy + y^2}$$

$$2) x^3 \ln x + \ln(4\sqrt{x} + 5)$$

3) Вычислить высоты треугольника со сторонами a , b , c .

Вариант 5

$$1) \quad x - \frac{x^3}{3} + \frac{x^5}{5}$$

$$2) \quad \frac{3 + e^{y-1}}{1 + x^2 |y - \operatorname{tg} z|}$$

Тема 3. Операции, выражения в языке Python. Решение задач с помощью условных операторов.

Лабораторно-практическая работа 2. Решение задач с помощью условных операторов.

Вариант 1

$$f(x) = \begin{cases} x^2 - 3x + 9 & \text{Если } x \leq 3; \\ \frac{1}{x^3 + 1} & \text{В противном случае.} \end{cases}$$

2. Даны три действительные числа. Возвести в квадрат те из них, значения которых неотрицательны, и в четвертую степень – отрицательные.

3. Найти $\max \{ \min(a, b), \min(c, d) \}$.

Вариант 2

$$f(x) = \begin{cases} -x + 3x + 9 & \text{Если } x \geq 3; \\ \frac{x}{x^3 - 1} & \text{В противном случае.} \end{cases}$$

2. Даны две точки $A(x_1, y_1)$, $B(x_2, y_2)$. Составить алгоритм, определяющий, которая из точек находится ближе к началу координат.

3. Даны три числа m, n, p . Определить какое из них равно d . Если ни одно не равно d , то найти $\max(d-m, d-n, d-p)$.

Вариант 3

$$f(x) = \begin{cases} 9 & \text{Если } x \leq -3 \\ \frac{1}{x^2 + 1} & \text{В противном случае.} \end{cases}$$

2. Даны два угла треугольника (в градусах). Определить существует ли такой треугольник. Если да, то будет ли он прямоугольным.

3. Даны четыре точки $A_1(x_1, y_1)$, $A_2(x_2, y_2)$, $A_3(x_3, y_3)$, $A_4(x_4, y_4)$. Определить будут ли они вершинами параллелограмма.

Вариант 4

$$f(x) = \begin{cases} 0 & \text{Если } x \leq 1 \\ \frac{1}{x + 6} & \text{В противном случае} \end{cases}$$

2. Даны действительные числа x и y , не равные друг другу. Меньшее из этих чисел заменить их полусуммой, а большее – их удвоенным произведением.

3. Даны три точки $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$. Определить будут ли они расположены на одной прямой. Если нет, то вычислить угол ABC .

Вариант 5

$$f(x) = \begin{cases} -3x + 9 & \text{Если } x \leq 7 \\ \frac{1}{x - 7} & \text{В противном случае} \end{cases}$$

2. На плоскости XOY задана своими координатами точка A. Указать, где она расположена: на какой оси или в каком координатном угле.

3. Даны действительные числа a, b, c. Удвоить эти числа, если $a \geq b \geq c$ и заменить их абсолютными значениями, если это не так.

Тема 4. Решение задач с помощью циклов.

Лабораторно-практическая работа 3. Решение задач с помощью циклов.

Вариант 1.

1. Имеется серия измерений элементов треугольника. Группы элементов пронумерованы. В серии в произвольном порядке могут встречаться такие группы элементов треугольника:

- 1) основание и высота;
- 2) две стороны и угол между ними (угол задан в радианах);
- 3) три стороны.

Составить программу, которая запрашивает номер группы элементов, вводит соответствующие элементы и вычисляет площадь треугольника. Вычисления прекратить, если в качестве номера группы введен 0.

2. Дано натуральное число N. Вычислить:

$$S = 1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \dots + (-1)^n \cdot \frac{1}{2^n}.$$

3. Найти сумму ряда с точностью $\varepsilon = 0,001$, общий член которого: $a_n = \frac{(-1)^{n-1}}{n^n};$

Вариант 2.

1. Начав тренировки, спортсмен в первый день пробежал 10 км. Каждый день он увеличивал дневную норму на 10% нормы предыдущего дня. Какой суммарный путь пробежит спортсмен за 7 дней?

2. Дано натуральное число N. Вычислить:

$$S = \frac{1}{\sin 1} + \frac{1}{\sin 1 + \sin 2} + \dots + \frac{1}{\sin 1 + \sin 2 + \dots + \sin N}.$$

3. Найти сумму ряда с точностью $\varepsilon = 0,001$, общий член которого:

$$a_n = \frac{1}{2^n} + \frac{1}{3^n};$$

Вариант 3.

1. Одноклеточная амeba каждые 3 часа делится на 2 клетки. Определить, сколько амeb будет через 3, 6, 9, 12, ..., 24 часа?

2. Дано натуральное число N. Вычислить произведение первых N сомножителей:

$$P = \frac{2}{3} \cdot \frac{4}{5} \cdot \frac{6}{7} \cdot \dots \cdot \frac{2 \cdot N}{2 \cdot N + 1}.$$

3. Найти сумму ряда с точностью $\varepsilon = 10^{-3}$, общий член которого:

$$a_n = \frac{2n-1}{2^n};$$

Вариант 4.

1. Около стены наклонно стоит палка длиной x м. Один ее конец находится на расстоянии y м от стены. Определить значение угла α между палкой и полом для значений $x = k$ м и y, изменяющегося от 2 до 3 м с шагом h м.

2. Дано натуральное число N. Вычислить:

$$\frac{\cos 1}{\sin 1} \times \frac{\cos 1 + \cos 2}{\sin 1 + \sin 2} \times \dots \times \frac{\cos 1 + \cos 2 + \dots + \cos N}{\sin 1 + \sin 2 + \dots + \sin N}.$$

3. Найти сумму ряда с точностью $\varepsilon = 0,001$, общий член которого:

$$a_n = \frac{1}{(3n-2)(3n+1)};$$

Вариант 5.

1. У гусей и кроликов вместе 64 лапы. Сколько могло быть кроликов и гусей (указать все сочетания, которые возможны)?

2. Дано действительное число. Вычислить:

$$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \frac{x^{13}}{13!}.$$

3. Найти сумму ряда с точностью $\varepsilon=10^{-3}$, общий член которого:

$$a_n = \frac{10^n}{n!};$$

Тема 5. Одномерные массивы. Двумерные массивы.

Лабораторно-практическая работа 4. Одномерные массивы. Двумерные массивы. Вставка, удаление и перестановка элементов.

Варианты 1-го задания: Задачи на одномерные массивы

1. Найти сумму всех чётных элементов массива, стоящих на чётных местах, то есть имеющие чётные номера.
2. Найти сумму и произведение первых пяти элементов массива.
3. Найти сумму элементов с k1-го по k2-ой, где k1 и k2 вводятся с клавиатуры. Сделайте проверку корректности их ввода.
4. Найти сумму элементов, больших данного числа А (А вводится с клавиатуры).
5. Найти сумму элементов, принадлежащих промежутку от А до В (А и В вводятся с клавиатуры).

Варианты 2-го задания: задачи на двумерные массивы

1. В прямоугольной матрице, размерностью n*m, заполненной случайным образом числами из промежутка [-40,30], найти сумму и количество элементов каждого столбца с заданным условием (хранить эти значения в массивах): элементы, кратные k1 или k2.
2. В прямоугольной матрице, размерностью n*m, заполненной случайным образом числами из промежутка [-40,30], найти сумму и количество элементов каждого столбца с заданным условием (хранить эти значения в массивах): элементы, попадающие в промежуток от А до В.
3. В прямоугольной матрице, размерностью n*m, заполненной случайным образом числами из промежутка [-40,30], найти сумму и количество элементов каждого столбца с заданным условием (хранить эти значения в массивах): данные элементы положительные и лежат выше главной диагонали.
4. В прямоугольной матрице, размерностью n*m, заполненной случайным образом числами из промежутка [-40,30], найти сумму и количество элементов каждого столбца с заданным условием (хранить эти значения в массивах): элементы, которые являются простыми числами.
5. Найти сумму элементов в строках с k1-й по k2-ю.

Варианты 3-го задания: Вставка, удаление и перестановка элементов

1. Вставить первую строку после строки, в которой находится первый встреченный максимальный элемент.
2. Вставить второй столбец после первого столбца, в котором все элементы положительны. Если такого столбца нет, то сообщить об этом.
3. Вставить нулевую строку и нулевой столбец перед строкой и столбцом, где находится первый минимальный элемент.
4. Вставить после строки, в которой есть заданное число А, последнюю строку.

5. Вставить перед столбцом, в котором нет отрицательных элементов, второй столбец.

Тема 6. Строковые данные.

Лабораторно-практическая работа 5. Строковый тип данных.

Для выполнения лабораторной работы № 4 необходимо составить три программы. Номер варианта – есть номер студента в группе. Возьмите из каждого блока задачу своего варианта. Таких блоков – три. Поэтому и задач у Вас – три. Рекомендации по разработке алгоритмов, использующих строки типа **String**.

1. Блок вариантов первой задачи по теме «Строки»

1. Подсчитать, сколько раз в данной строке встречается буква *Ch*, вводимая с клавиатуры.
2. Дан текст, в котором записано одно из стихотворений А.С. Пушкина. Сколько раз в каждой строке встречается данный символ?
3. Дан текст, в котором записано одно из стихотворений А.С. Пушкина. Сколько раз встречаются гласные буквы?
4. Из заданной строки удалить среднюю букву, если длина строки нечетная, иначе — удалить две средние буквы.
5. Заменить в заданной строке все буквы *Ch1* на *Ch2* (их значения вводить с клавиатуры).

2. Блок вариантов второй задачи по теме «Строки»

1. Дана символьная строка, заканчивающаяся точкой. Подсчитать, сколько слов в строке.
2. Дана символьная строка, содержащая английский текст. Найти количество слов, начинающихся с буквы *b*.
3. Дана символьная строка. Подсчитать в ней количество вхождений букв *r*, *k*, *t*.
4. Дана символьная строка. Определить, сколько в ней знаков «*», «;», «:».
5. Дана символьная строка, содержащая текст, который заканчивается точкой. Найти длину самого короткого слова и самого длинного слова.

3. Блок вариантов третьей задачи по теме «Строки»

1. Символьная строка содержит одно слово. Проверить, будет ли оно читаться одинаково справа налево и слева направо (т.е. является ли оно палиндромом).
2. В записке слова зашифрованы – каждое из них записано наоборот. Расшифровать сообщение.
3. Проверить, одинаковое ли число открывающихся и закрывающихся скобок в данной строке.
4. Символьная строка, содержащая произвольный русский текст, состоит не более, чем из 200 символов. Написать, какие буквы и сколько раз встречаются в этом тексте. Ответ должен приводиться в грамматически правильной форме, например: *a* – 25 раз, *k* – 3 раза.
5. Упорядочить данный массив английских слов по алфавиту, рассматривая только первый символ каждого слова.

Вопросы к контрольной работе 1

1. Особенности языка Python.
2. Структура программы в языке Python.
3. Понятие переменной.
4. Типы данных.
5. Инициализация переменных.
6. Основные типы переменных.
7. Понятие константы. Объявление. Инициализация.

8. Простейшие арифметические операции.
9. Принципы структурного программирования.
10. Оператор множественного ветвления switch.
11. Условный оператор if-else.
12. Вложенный оператор if.
13. Операторы цикла.
14. Одномерные массивы: объявление, инициализация, доступ к элементам.
15. Двумерные массивы: объявление, инициализация, доступ к элементам.
16. Доступ к элементам одномерного массива.
17. Доступ к элементам двумерного массива.
18. Стандартный класс string. Основные понятия.
19. Основные операции класса string.

Тема 7. Функции.

Лабораторно-практическая работа 6. Процедуры и функции.

Варианты 1-го задания: Программирование процедур

1. Дано натуральное число N . Составить процедуру (функцию) формирования массива, элементами которого являются цифры числа N .
2. Составить процедуру (функцию), определяющую, в каком из данных двух чисел больше цифр.
3. Заменить данное натуральное число на число, которое получается из исходного записью его цифр в обратном порядке. Например, дано число 156, нужно получить 651.
4. Даны натуральные числа K и N . Составить процедуру (функцию) формирования массива A , элементами которого являются числа, сумма цифр которых равна K и которые не больше N .
5. Даны три квадратных матрицы A, B, C n -го порядка. Вывести на печать ту из них, норма которой наименьшая. Пояснение. Нормой матрицы считать максимум из абсолютных величин ее элементов.

Варианты 2-го задания: Программирование функций

1. Треугольник задан координатами своих вершин. Составить функцию вычисления его площади.
2. Составить функцию нахождения наибольшего общего делителя и наименьшего общего кратного двух натуральных чисел ($НОК(A, B) = A * B / НОД(A, B)$).
3. Составить функцию нахождения наибольшего общего делителя четырех натуральных чисел.
4. Составить функцию нахождения наименьшего общего кратного трех натуральных чисел.
5. Написать функцию нахождения суммы большего и меньшего из 3 чисел.

Лабораторно-практическая работа 7. Структуры.

Вариант 1.

1. Описать структуру с именем STUDENT, содержащую следующие поля:
 - NAME – фамилия и инициалы;
 - GROUP – номер группы;
 - SES – успеваемость (массив из пяти элементов).
2. Написать программу, выполняющую следующие действия :
 - ввод с клавиатуры данных в массив STUD1, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по возрастанию содержимого поля GROUP;
 - вывод на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4,0;
 - если таких нет, вывести соответствующее сообщение.

Вариант 2

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

- NAME - фамилия и инициалы;
- GROUP- номер группы;
- SES- успеваемость (массив из пяти элементов).

2. Написать программу, выполняющую следующие действия :

- ввод с клавиатуры данных в массив STUD1, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по возрастанию среднего балла;
- вывод на дисплей фамилий и номеров групп для всех студентов, имеющих оценки 4 и 5;
- если таких нет, вывести соответствующее сообщение.

Вариант 3

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

- NAME - фамилия и инициалы;
- GROUP- номер группы;
- SES- успеваемость (массив из пяти элементов).

2. Написать программу, выполняющую следующие действия :

- ввод с клавиатуры данных в массив STUD1, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по алфавиту;
- вывод на дисплей фамилий и номеров групп для всех студентов, имеющих хотя бы одну оценку 2;
- если таких студентов нет, вывести соответствующее сообщение.

Вариант 4

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

- NAZN — название пункта назначения рейса;
- NUMR — номер рейса;
- TIP — тип самолета.

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив AIRPORT, состоящий из семи элементов типа AEROFLOT; записи должны быть упорядочены по возрастанию номера рейса;
- вывод на экран номеров рейсов и типов самолетов, вылетающих в пункт назначения, название которого совпало с названием, введенным с клавиатуры;
- если таких рейсов нет, выдать на дисплей соответствующее сообщение.

Вариант 5

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

- NAZN — название пункта назначения рейса;
- NUMR — номер рейса;
- TIP — тип самолета.

2. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив AIRPORT, состоящий из семи элементов типа AEROFLOT; записи должны быть размещены в алфавитном порядке по названиям пунктов назначения;
- вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры;
- если таких рейсов нет, выдать на дисплей соответствующее сообщение.

Тема 8. Рекурсивные функции.

Лабораторно-практическая работа 8. Рекурсивные функции.

Варианты задания:

1. Найдите сумму цифр заданного натурального числа, используя рекурсивную подпрограмму.
2. Подсчитать количество цифр в заданном натуральном числе, используя рекурсивную подпрограмму.

3. Описать функцию $C(m, n)$, где $0 \leq m \leq n$, для вычисления биномиального коэффициента C_n^m по следующей формуле:

$$C_n^0 = C_n^n = 1; C_n^m = C_{n-1}^m + C_{n-1}^{m-1} \text{ при } 0 < m < n$$

используя рекурсивную подпрограмму

4. Описать рекурсивную функцию $\text{Root}(f, b, \varepsilon)$, которая методом деления отрезка пополам находит с точностью ε корень уравнения $f(x) = 0$ на отрезке $[a, b]$ (считать, что $\varepsilon > 0$, $a < b$, $f(a) \cdot f(b) < 0$ и $f(x)$ - непрерывная и монотонная на отрезке $[a, b]$).
5. Описать функцию $\text{min}(x)$ для определения минимального элемента линейного массива x , введя вспомогательную рекурсивную функцию $\text{min1}(k)$, находящую минимум среди последних элементов массива x , начиная с k -го.

Тема 9. Работа с файлами.

Лабораторно-практическая работа 9. Работа с файлами.

Вариант №1

1. Заполнить файл f целыми числами, полученными с помощью генератора случайных чисел. Получить в файле g те компоненты файла f , которые являются четными. Багаж пассажира характеризуется количеством вещей и общим весом вещей.

2. Дан файл **Bagazh**, содержащий сведения о багаже нескольких пассажиров. Сведения о багаже каждого пассажира представляют собой запись с двумя полями: одно поле целого типа (количество вещей) и одно — действительное (вес в килограммах). Найти багаж, средний вес одной вещи в котором отличается не более чем на 0,3 кг от общего среднего веса одной вещи.

3. Дан файл, содержащий текст, записанный строчными русскими буквами. Получить в другом файле тот же текст, записанный заглавными буквами.

Вариант №2

1. Записать в файл N действительных чисел. Вычислить произведение компонент файла и вывести на печать.

2. Багаж пассажира характеризуется количеством вещей и общим весом вещей. Дан файл **Bagazh**, содержащий сведения о багаже нескольких пассажиров. Сведения о багаже каждого пассажира представляют собой запись с двумя полями: одно поле целого типа (количество вещей) и одно — действительное (вес в килограммах). Найти число пассажиров, имеющих более двух вещей и число пассажиров, количество вещей которых превосходит среднее число вещей.

3. Дан файл, содержащий произвольный текст. Выяснить, чего в нем больше: русских букв или цифр.

Вариант №3

1. Заполнить файл f целыми числами, полученными с помощью генератора случайных чисел. Получить в файле g все компоненты файла f , которые делятся на m и не делятся на n .

2. Багаж пассажира характеризуется количеством вещей и общим весом вещей. Дан файл **Bagazh**, содержащий сведения о багаже нескольких пассажиров. Сведения о багаже каждого пассажира представляют собой запись с двумя полями: одно поле целого типа (количество вещей) и одно - действительное (вес в килограммах). Выяснить, имеется ли пассажир, багаж которого состоит из одной вещи весом менее 30 кг.

3. Дан файл, содержащий текст на русском языке. Выяснить, входит ли данное слово в указанный текст, и если да, то сколько раз.

Вариант №4

1. Записать в файл N целых чисел, полученных с помощью генератора случайных чисел. Подсчитать количество пар противоположных чисел среди компонент этого файла.
2. Дан файл **Bibl**, содержащий сведения о книгах. Сведения о каждой из книг — это фамилия автора, название и год издания. Найти название книг данного автора, изданных с 1960 года.
3. Дан файл, содержащий текст на русском языке. В предложениях некоторые из слов записаны подряд несколько раз (предложение заканчивается точкой или знаком восклицания). Получить в новом файле отредактированный текст, в котором удалены повторные вхождения слов в предложение.

Вариант №5

1. Заполнить файл f целыми числами, полученными с помощью генератора случайных чисел. Из файла f получить файл g , исключив повторные вхождения чисел. Вывести файл g на печать.
2. Дан файл **Bibl**, содержащий сведения о книгах. Сведения о каждой из книг — это фамилия автора, название и год издания. Определить, имеется ли книга с названием "Информатика". Если да, то напечатать фамилию автора и год издания. Если таких книг несколько, то напечатать имеющиеся сведения обо всех этих книгах.
3. Дан файл, содержащий текст, набранный заглавными русскими буквами. В нем содержатся также знаки препинания. Провести частотный анализ текста, т.е. указать (в процентах) сколько раз встречается та или иная буква.

Лабораторно-практическая работа 10. Динамические переменные.

Вариант №1

1. Запишите программу с определением константы $Const X: Longint = 2147483000$ и выводом соответствующих ей: значения, адреса динамической памяти.
2. С помощью динамических переменных реализуйте вещественную матрицу размером 200×100 . Каждому элементу матрицы присвойте случайное значение из заданного диапазона. Отсортируйте элементы в строках и строки по значению первого элемента. Выдайте на экран первую подматрицу размером 20×10 .

Вариант №2

1. Опишите матрицу A размером 100×200 с элементами типа *Char*. Заполните ее путем обращения к процедуре: *FillChar(A, 20000, 'X')*.
2. Составьте программу размещения в ДРП, строка за строкой, треугольной матрицы размером N , в i -той строке которой i компонентов типа *Word*, $i = 1, 2, 3, \dots, N$. Занесите в элемент массива *MemW*, соответствующий элементу $A^{[1]}_{[1]}$, значение «1», а в следующие девять — значения 2, 3, ..., 9. Выведите элементы первых трех строк матрицы и объясните полученную картину.

Вариант №3

1. Запишите программу, которая выводит значения адреса сегмента ее кода, начала данных из кучи и стека. Сравните результаты ее запуска из оболочки Turbo Pascal и из операционной системы.
2. С помощью динамических переменных реализуйте вещественную матрицу размером 200×100 . Каждому элементу матрицы присвойте случайное значение из заданного диапазона. Отсортируйте элементы в строках и строки по значению первого элемента. Выдайте на экран первую подматрицу размером 20×10 .

Вариант №4

1. Создайте динамическую переменную, которая является записью с фамилией, годом рождения, ростом и весом человека. Соответственно задайте значения полей: Иванов, 1939, 174,72. Затем выведите содержимое этой записи на экран дисплея.
2. С помощью динамических переменных реализуйте вещественную матрицу размером 100 x 200. Каждому элементу матрицы присвойте случайное значение из заданного диапазона. Отсортируйте элементы массива по сквозному порядку их размещения в матрице как в одномерном массиве. Выдайте на экран первую подматрицу размером 10 x 20. Для сортировки используйте метод пузырька.

Вариант №5

1. Составьте программу, в которой 8-10 переменных разной длины, например, строки символов, размещаются в ДРП, причем размещение чередуется с освобождением памяти так, что возникает фрагментация. Переменные меньшей длины размещайте после освобождения памяти 1-2 переменных большей длины. Используя функции *Ofs*, *Seg*, получите карту распределения памяти на промежуточных и окончательных этапах.
С помощью динамических переменных реализуйте вещественную матрицу размером $n \times m$ максимального размера на всю динамическую память. Каждому элементу матрицы присвойте случайное значение из заданного диапазона. Отсортируйте элементы массива по сквозному порядку их размещения в матрице как в одномерном массиве. Выдайте на экран первую подматрицу размером 20 x 10. Для сортировки используйте метод обменной сортировки.

Лабораторно-практическая работа 11. Односвязный список.

Вариант 1.

1. Описать структуру с именем STUDENT, содержащую следующие поля:
 - NAME – фамилия и инициалы;
 - GROUP – номер группы;
 - SES - успеваемость (массив из пяти элементов).
2. Написать программу, выполняющую следующие действия:
 - ввод с клавиатуры данных в список STUD1, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по возрастанию содержимого поля GROUP;
 - вывод на дисплей фамилий и номеров групп для всех студентов, включенных в список, если средний балл студента больше 4,0;
 - если таких нет, вывести соответствующее сообщение.

Вариант 2

1. Описать структуру с именем STUDENT, содержащую следующие поля:
 - NAME - фамилия и инициалы;
 - GROUP- номер группы;
 - SES- успеваемость (массив из пяти элементов).
2. Написать программу, выполняющую следующие действия:
 - ввод с клавиатуры данных в список STUD1, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по возрастанию среднего балла;
 - вывод на дисплей фамилий и номеров групп для всех студентов, имеющих оценки 4 и 5;
 - если таких нет, вывести соответствующее сообщение.

Вариант 3

1. Описать структуру с именем STUDENT, содержащую следующие поля:
 - NAME - фамилия и инициалы;

- GROUP- номер группы;
 - SES- успеваемость (массив из пяти элементов).
2. Написать программу, выполняющую следующие действия:
- ввод с клавиатуры данных в список STUD1, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по алфавиту;
 - вывод на дисплей фамилий и номеров групп для всех студентов, имеющих хотя бы одну оценку 2;
 - если таких студентов нет, вывести соответствующее сообщение.

Вариант 4

1. Описать структуру с именем AEROFLOT, содержащую следующие поля:
- NAZN — название пункта назначения рейса;
 - NUMR — номер рейса;
 - TIP — тип самолета.
2. Написать программу, выполняющую следующие действия:
- ввод с клавиатуры данных в список AIRPORT, состоящий из семи элементов типа AEROFLOT; записи должны быть упорядочены по возрастанию номера рейса;
 - вывод на экран номеров рейсов и типов самолетов, вылетающих в пункт назначения, название которого совпало с названием, введенным с клавиатуры;
 - если таких рейсов нет, выдать на дисплей соответствующее сообщение.

Вариант 5

1. Описать структуру с именем AEROFLOT, содержащую следующие поля:
- NAZN — название пункта назначения рейса;
 - NUMR — номер рейса;
 - TIP — тип самолета.
2. Написать программу, выполняющую следующие действия:
- ввод с клавиатуры данных в список AIRPORT, состоящий из семи элементов типа AEROFLOT; записи должны быть размещены в алфавитном порядке по названиям пунктов назначения;
 - вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры;
 - если таких рейсов нет, выдать на дисплей соответствующее сообщение.

Вопросы к контрольной работе 2

1. Описание массива. Объявление и инициализация массива.
2. Описание строки. Объявление и инициализация строки.
3. Массив. Прямой доступ к элементам массива.
4. Стандартные методы обработки строк.
5. Принципы модульного программирования.
6. Понятие функции. Стандартные функции.
7. Описание функции.
8. Инструкция def.
9. Вызов функции.
10. Передача параметров в функцию.
11. Передача параметров по адресу.
12. Возврат значений из функций.
13. Анонимные функции.
14. Инструкция lambda.
15. Запись данных в файл.
16. Чтение данных из файла.

17. Режимы доступа к файлу.

Тема 10. Основы объектно-ориентированного программирования.

Тема 11. Основные понятия объектно-ориентированного программирования (ООП).

Лабораторно-практическая работа 12. Реализация классов.

Цели:

Изучение структуры класса, механизм создания и использования, описание членов-данных класса и методов доступа к ним. При реализации классов поля должны быть скрытыми.

Задание:

- Реализовать пользовательский класс в соответствии с вариантом задания.
- Добавить в класс необходимые поля и методы.
- Определить метод установки свойств (при недопустимых аргументах функции возвращать «false» и выдавать текст ошибки на экран).
- Определить метод чтения свойств.
- Написать демонстрационную программу, в которой показать использование объектов созданного класса.

Варианты заданий:

Вариант 1

Класс Треугольник

Свойства: три стороны

Операции:

- увеличение/уменьшение размера сторон заданное количество раз;
- вычисление периметра;
- вычисление площади;
- определение значений углов.

Вариант 2

Класс Треугольник

Свойства: три стороны

Операции:

- вычисление средней линии для любой из сторон;
- определение вида треугольника по величине углов (Остроугольный, Тупоугольный, Прямоугольный);
- определение значений углов.

Вариант 3

Класс Треугольник

Свойства: две стороны и угол между ними

Операции:

- увеличение/уменьшение угла на заданное количество процентов;
- определение вида треугольника по числу равных сторон (Разносторонний, Равнобедренный, Равносторонний);
- определение расстояния между центрами вписанной и описанной окружностей;
- определение значений углов.

Вариант 4

Класс Треугольник

Свойства: две стороны и угол между ними

Операции:

- увеличение/уменьшение угла в заданное количество раз;
- вычисление длины биссектрисы, принадлежащей любому углу;
- вычисление длин отрезков, на которые биссектриса делит любую сторону;
- определение значений углов.

Вариант 5

Класс Треугольник

Свойства: сторона и два прилежащих к ней угла

Операции:

- увеличение/уменьшение размера стороны (из свойств) в заданное количество раз;
- вычисление длины медианы, принадлежащей любой стороне;
- определение подобен ли другой треугольник данному;
- определение значений сторон.

Тема 12. Классы.

Лабораторно-практическая работа 13. Конструкторы и деструкторы.

Цели:

Изучение возможности инициализации объектов класса с помощью конструкторов и уничтожение их с помощью деструкторов.

Задание:

Пользовательский класс `MyString` должен содержать необходимые элементы-данные.

– Конструкторы (без параметров, с параметрами, копирования) для создания строк: `MyString (...)`;

– Деструктор: `~MyString()`;

– Метод ввода исходной строки: `set()`;

– Метод изменения исходной строки согласно варианту;

– Метод вывода на экран: `print(...)`;

– Каждый вызов методов (в том числе конструкторов и деструктора)

сопровождается выдачей соответствующего сообщения;

– Код методов – вне пространства определения класса.

– Написать демонстрационную программу, в которой показать использование объектов созданного класса.

Варианты заданий:

Варианты:

1. Длина L нечетная, то удаляется символ, стоящий посередине строки;
2. Длина L четная, то удаляются 2 первых и 2 последних символа;
3. Длина L кратна 2-м, то удаляются все цифры, которые делятся на 2;
4. Длина L кратна 3-м, то удаляются все цифры, делящиеся на 3;
5. Длина $L > 10$, то удаляются все цифры.

Вопросы к контрольной работе 3

1. Принципы объектно-ориентированного программирования. Инкапсуляция.
2. Принципы объектно-ориентированного программирования. Полиморфизм.
3. Принципы объектно-ориентированного программирования. Наследование.
4. Классы. Объявление открытых переменных.
5. Классы. Инициализация открытых переменных.
6. Классы. Объявление закрытых переменных.

7. Классы. Инициализация закрытых переменных.
8. Спецификаторы доступа.
9. Функции-члены класса.
10. Связывание функций.
11. Встроенные функции.
12. Конструкторы. Основные понятия.
13. Конструкторы по умолчанию.
14. Конструкторы с параметрами.
15. Область действия класса. Управление доступом к членам класса.
16. Отделение интерфейса от реализации. Понятие интерфейса.
17. Понятие конструктора. Использование конструктора с аргументами по умолчанию. Конструктор по умолчанию.
18. Понятие деструктора. Когда вызываются конструкторы и деструкторы.
19. Константные объекты и функции-члены.
20. Дружественные функции и дружественные классы.
21. Использование указателя this.
22. Операции new и delete.
23. Статические члены класса.

Тема 13. Поля и методы класса.

Лабораторно-практическая работа 14. Статические, константные и дружественные члены класса

Цели:

Изучить работу со статическими и константными членами класса.

Задание:

Для задания практической работы 3 «Перегрузка операций» выполнить доработку программы в соответствии с заданиями:

ЗАДАНИЕ 1. Статические члены-данные класса

1. Добавить статическое поле `int count`, выступающее в роли счетчика объектов класса.
2. Деструктор класса должен уменьшать на единицу значение счетчика.
3. Добавить статический метод `int getCount()` возвращающий значение счетчика.
4. Продемонстрировать изменение значения статического поля.

ЗАДАНИЕ 2. Константные методы и объекты

1. Определить какие методы являются константными, определить константные параметры и константные возвращаемые значения методов.
2. Добавить константное поле, хранящее идентификатор объекта (номер созданного объекта), предусмотреть методы вывода информации о идентификаторе.
3. Описать и инициализировать обычные и константные объекты.
4. Выполнить вызовы обычных и константных методов для каждого вида объектов.
5. Провести тестирование программы: Откомпилировать программу. Имеются ли ошибки компиляции и какие? Если имеются, то закомментировать соответствующие строки кода и вновь провести компиляцию. Какие предупреждения выдает компилятор и в чем их смысл? Как их можно объяснить с позиции обеспечения надежности программы?

Тема 14. Перегрузка операторов.

Лабораторно-практическая работа 15. Перегрузка операций.

Цели:

Изучить механизм перегрузки операций.

Задание:

Все классы следует наделить конструкторами, деструктором. Необходимо явно реализовать конструктор копирования и перегрузить оператор присваивания. Необходимо подготовить демонстрацию по работе перегруженных для класса операторов. *Если в задании требуется перегрузить более двух операций, то можно выбрать любые две операции.*

Варианты заданий:**Варианты:**

1. Создать класс ПРЯМОУГОЛЬНИК со сторонами параллельными осям координат (прямоугольная система координат ОХУ). Реализовать метод вывода на экран информации о прямоугольнике. Перегрузить бинарный оператор разности двух прямоугольников (-); унарный оператор (-): симметричное отображение прямоугольника относительно оси координат ОХ и ОУ. Следует учесть, что результатом выполнения оператора может быть пустой прямоугольник.
2. Создать класс, описывающий тип ВРЕМЯ. Класс должен включать в себя атрибуты, описывающие часы, минуты, секунды и миллисекунды и иметь метод для вывода времени на экран. Для данного класса перегрузить следующие бинарные операторы: суммы(+), разности (-).
3. Создать класс МНОГОЧЛЕН степени n от одной переменной x , задаваемый массивом своих коэффициентов (массив должен задаваться внутри конструктора, используя датчик случайных чисел). Класс должен включать конструктор, которому в качестве параметра передается степень многочлена; деструктор; конструктор копирования, метод, который печатает уравнение на экран. Для данного класса перегрузить следующие бинарные операторы: суммы двух многочленов (+), разности двух многочленов (-), операцию присваивания (=).
4. Создать класс ОТРЕЗОК ЧИСЛОВОЙ ПРЯМОЙ (сегмент). Предусмотреть несколько конструкторов для создания объектов класса, реализовать метод вывода отрезка на экран. Перегрузить следующие бинарные операторы: дополнение одного сегмента другим (+), пересечение двух сегментов (*), несимметрическая разность сегментов (-) $\{[1,5]-[3,6]=[1,3]\}$. Следует учесть, что результатом выполнения оператора может быть пустой сегмент, в этом случае следует вернуть отрезок нулевой длины $[0,0]$.
5. Создать класс КВАДРАТНАЯ МАТРИЦА 3×3 . Элементы матрицы следует задавать внутри конструктора, используя датчик случайных чисел. Класс должен включать конструктор; деструктор; конструктор копирования; метод для вывода матрицы на экран. Перегрузить следующие бинарные операторы: сумма матриц (+), произведение матриц (*).

Тема 15. Наследование.**Лабораторно-практическая работа 16. Наследование. Управление доступом к членам базового класса.****Цели:**

Изучить возможности наследования классов на языке C++.

Задание:

Реализовать иерархию классов для простого (задание 1) и множественного (задание 2) наследования:

Варианты заданий:**Задание 1:**

1. Создать класс квадрат, члены класса – длина стороны. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – диагональ, периметр, площадь. Создать производный класс – правильная квадратная призма с высотой H , добавить в класс метод определения объема фигуры, перегрузить методы расчета площади и вывода сведений о фигуре. Написать программу, демонстрирующую работу с этими классами: дано N квадратов и M призм, найти квадрат с максимальной площадью и призму с максимальной диагональю.

2. Создать класс треугольник, члены класса – длины 3-х сторон. Предусмотреть в классе методы проверки существования треугольника, вычисления и вывода сведений о фигуре – длины сторон, углы, периметр, площадь. Создать производный класс – равносторонний треугольник, перегрузить в классе проверку, является ли треугольник равносторонним и метод вывода сведений о фигуре. Написать программу, демонстрирующую работу с классом: дано K треугольников и L равносторонних треугольников, найти среднюю площадь для K треугольников и наибольший равносторонний треугольник.

3. Создать класс окружность, член класса – радиус R . Предусмотреть в классе методы вычисления и вывода сведений о фигуре – площади, длины окружности. Создать производный класс – круглый прямой цилиндр с высотой h , добавить в класс метод определения объема фигуры, перегрузить методы расчета площади и вывода сведений о фигуре. Написать программу, демонстрирующую работу с классом: дано N окружностей и M цилиндров, найти окружность максимальной площади и средний объем цилиндров.

4. Создать класс квадрат, члены класса – длина стороны. Предусмотреть в классе методы вычисления и вывода сведений о фигуре – диагоналей, периметр, площадь. Создать производный класс – правильная пирамида с апофемой h , добавить в класс метод определения объема фигуры, перегрузить методы расчета площади и вывода сведений о фигуре. Написать программу, демонстрирующую работу с классом: дано N квадратов и M пирамид, найти квадрат с минимальной площадью и количество пирамид с высотой более числа a (a вводить).

5. Создать класс четырехугольник, члены класса – координаты 4-х точек. Предусмотреть в классе методы проверки существования четырехугольника вычисления и вывода сведений о фигуре – длины сторон, диагоналей, периметр, площадь. Создать производный класс – параллелограмм, предусмотреть в классе проверку, является ли фигура параллелограммом. Написать программу, демонстрирующую работу с классом: дано N четырехугольников и M параллелограммов, найти среднюю площадь N четырехугольников и параллелограммы наименьшей и наибольшей площади.

Тема 16. Виртуальные функции. Абстрактные классы.

Лабораторно-практическая работа 17. Программирование динамических, виртуальных методов.

Цели:

Изучить возможности виртуальных функций при наследовании классов на языке C++, абстрактные классы.

Задание:

1. Определить иерархию классов (в соответствии с вариантом).
2. Определить в классе статическую компоненту - указатель на начало связанного списка объектов и статическую функцию для просмотра списка.
3. Реализовать классы.
4. Написать демонстрационную программу, в которой создаются объекты различных классов и помещаются в список, после чего список просматривается.
5. Сделать соответствующие методы не виртуальными и посмотреть, что будет.
6. Реализовать вариант, когда объект добавляется в список при создании, т.е. в конструкторе.

Методические указания

1. Для определения иерархии классов связать отношением наследования классы заданного варианта. Из перечисленных классов выбрать один, который будет стоять во главе иерархии. Это абстрактный класс.

2. Определить в классах все необходимые конструкторы и деструктор.

3. Поля класса специфицировать как `protected`.

4. Пример определения статических компонентов:

static person begin; // указатель на начало списка*

static void print(); // просмотр списка

5. Статическое поле инициализировать вне определения класса, в глобальной области.

6. Для добавления объекта в список предусмотреть метод класса, т.е. объект сам добавляет себя в список. Например, *a.Add()* - объект *a* добавляет себя в список.

Включение объекта в список можно выполнять при создании объекта, т.е. поместить операторы включения в конструктор. В случае иерархии классов, включение объекта в список должен выполнять только конструктор базового класса. Вы должны продемонстрировать оба этих способа.

7. Список просматривать путем вызова виртуального метода `Show` каждого объекта.

8. Статический метод просмотра списка вызывать не через объект, а через класс.

Варианты заданий:

Варианты:

- 1) студент, преподаватель, персона, завкафедрой;
- 2) служащий, персона, рабочий, инженер;
- 3) рабочий, кадры, инженер, администрация;
- 4) деталь, механизм, изделие, узел;
- 5) организация, страховая компания, судостроительная компания, завод.

Вопросы к контрольной работе 4

1. Перегрузка функций.
2. Шаблоны функций.
3. Понятие перегрузки операций.
4. Функции-операции как члены класса и как дружественные функции.
5. Перегрузка операций ввода-вывода в поток, бинарных и унарных операций.
6. Перегрузка `++` и `--`.
7. Понятие наследования. Механизм ограничения доступа при наследовании.
8. Приведение типа указателя базового класса к указателю производного класса и наоборот.
9. Переопределение членов базового класса в производном классе.
10. Конструкторы и деструкторы в производных классах.
11. Неявное преобразование объектов производных классов в объекты базового класса.
12. Множественное наследование.
13. Понятие виртуальной функции.
14. Абстрактные и конкретные базовые классы.
15. Статическое и динамическое связывание.
16. Виртуальные деструкторы.
17. Виртуальные конструкторы.

Тема 17. Исключительные ситуации. Классы исключений.

Лабораторно-практическая работа 18. Обработка исключительных ситуаций.

Цели:

Создание консольного приложения, состоящего из нескольких файлов; разработка программы, обрабатывающей исключительные ситуации.

Задание:

1. Реализовать класс, перегрузить для него операции, указанные в варианте.
2. Определить исключительные ситуации.
3. Предусмотреть генерацию исключительных ситуаций.

Варианты заданий:

Варианты:

1. Класс- контейнер ВЕКТОР с элементами типа `int`.

Реализовать операции:

- `[]` - доступа по индексу;
- `()` - определение размера вектора;
- `+` число - добавляет константу ко всем элементам вектора;
- `- n` - удаляет `n` элементов из конца вектора.

2. Класс- контейнер ВЕКТОР с элементами типа `int`.

Реализовать операции:

- `[]` - доступа по индексу;
- `int()` - определение размера вектора;
- `- n` - удаляет `n` элементов из конца вектора;
- `+ n` - добавляет `n` элементов в конец вектора.

3. Класс- контейнер ВЕКТОР с элементами типа `int`.

Реализовать операции:

- `[]` - доступа по индексу;
- `++` - добавляет элемент в вектор (постфиксная операция добавляет элемент в конец, префиксная в начало)

4. Класс- контейнер ВЕКТОР с элементами типа `int`.

Реализовать операции:

- `[]` - доступа по индексу;
- `()` - определение размера вектора;
- `--` - удаляет элемент из вектора (постфиксная операция удаляет элемент из конца вектора, префиксная - из начала)

5. Класс- контейнер ВЕКТОР с элементами типа `int`.

Реализовать операции:

- `[]` - доступа по индексу;
- `int()` - определение размера вектора;
- `*` вектор - умножение элементов векторов `a[i]*b[i]`;
- `+ n` - переход вправо к элементу с номером `n`.

Тема 18. Шаблоны классов.

Лабораторно-практическая работа 19. Применение шаблонов классов.

Цели:

Получить практические навыки создания шаблонов и использования их в программах C++.

Задание:

1. Создать шаблон заданного класса. Определить конструкторы, деструктор, перегруженную операцию присваивания ("`=`") и операции, заданные в варианте задания.
2. Определить пользовательский класс, который будет использоваться в качестве параметра шаблона. Определить в классе необходимые функции и перегруженные операции.

3. Написать программу, в которой демонстрируется использование шаблона для стандартного и пользовательского типа.

Методические указания:

1. Класс реализовать как динамический массив. Для этого определение класса должно иметь следующие поля:

- указатель на начало массива;
- максимальный размер массива;
- текущий размер массива.

2. Для ввода и вывода определить в классе функции `input` и `print`.

3. Чтобы у вас не возникало проблем, аккуратно работайте с константными объектами. Например:

- конструктор копирования следует определить так:

`MyTmp (const MyTmp& ob);`

- операцию присваивания перегрузить так:

`MyTmp& operator = (const MyTmp& ob);`

4. Для шаблонов множеств, списков, стеков и очередей в качестве стандартных типов использовать символьные, целые и вещественные типы. Для пользовательского типа взять класс из лабораторной работы № 1.

5. Для шаблонов массивов в качестве стандартных типов использовать целые и вещественные типы. Для пользовательского типа взять класс “комплексное число” `complex`.

```
class complex{  
    int re; // действительная часть  
    int im; // мнимая часть  
    public;  
    // необходимые функции и перегруженные операции  
};
```

6. Тестирование должно быть выполнено для всех типов данных и для всех операций.

Варианты заданий:

Варианты:

1. Класс - одномерный массив. Дополнительно перегрузить следующие операции:
[] - доступ по индексу;
== - проверка на равенство;
!= - проверка на неравенство.
2. Класс - множество `set`. Дополнительно перегрузить следующие операции:
+ - добавить элемент в множество (типа `set+item`);
+ - объединение множеств;
* - пересечение множеств;
3. Класс - множество `set`. Дополнительно перегрузить следующие операции:
+ - добавить элемент в множество (типа `item + set`);
+ - объединение множеств;
== - проверка множеств на равенство.
4. Класс `□` множество `set`. Дополнительно перегрузить следующие операции:
- - удалить элемент из множества (типа `set-item`);
* - пересечение множеств;
< - сравнение множеств.
5. Класс - множество `set`. Дополнительно перегрузить следующие операции:
- - удалить элемент из множества (типа `set-item`);
> - проверка на подмножество;

!= - проверка множеств на неравенство.

Вопросы к контрольной работе 5

1. Шаблоны классов.
2. Шаблоны классов и нетиповые параметры.
3. Шаблоны и статические члены.
4. Понятие исключения. Когда должна использоваться обработка исключений.
5. Генерация исключений. Повторная генерация исключений.
6. Перехватывание исключений.
7. Спецификация исключений.
8. Обработка неожиданных исключений.
9. «Раскручивание» стека. Иерархия исключений стандартной библиотеки.
10. Конструкторы, деструкторы и обработка исключений.
11. Исключения и наследование.
12. Обработка неуспешного выполнения new.

Критерии оценки лабораторных работ:

– оценка «отлично» выставляется обучающемуся, если студент продемонстрировал глубокие знания теоретического материала и умение их применять, обоснованно изложил свои мысли, сделал необходимые выводы;

– оценка «хорошо» выставляется обучающемуся, если студент продемонстрировал глубокие знания теоретического материала и умение их применять, обоснованно изложил свои мысли, сделал необходимые выводы, допущены некоторые неточности, имеется одна негрубая ошибка;

– оценка «удовлетворительно» выставляется обучающемуся, если студент ответил на вопросы преимущественно верно, имеются затруднения в формулировке выводов, имеются одна или две негрубые ошибки;

– оценка «неудовлетворительно» выставляется обучающемуся, если студент не дал ответы на поставленные вопросы, обоснования неверные, либо дан верный ответ без его обоснования, сделаны грубые ошибки, отсутствуют знания по основам делопроизводства.

Критерии оценки контрольных работ:

– оценка «отлично» выставляется обучающемуся, если студент продемонстрировал глубокие знания теоретического материала и умение их применять, обоснованно изложил свои мысли, сделал необходимые выводы и учел основные нормативно-правовые документы по информационной безопасности;

– оценка «хорошо» выставляется обучающемуся, если студент продемонстрировал глубокие знания теоретического материала и умение их применять, обоснованно изложил свои мысли, сделал необходимые выводы и учел основные нормативно-правовые документы по информационной безопасности, допущены некоторые неточности, имеется одна негрубая ошибка.

– оценка «удовлетворительно» выставляется обучающемуся, если студент ответил на вопросы преимущественно верно, имеются затруднения в формулировке выводов, имеются одна или две негрубые ошибки, учтены не все нормативно-правовые документы по информационной безопасности;

– оценка «неудовлетворительно» выставляется обучающемуся, если студент не дал ответы на поставленные вопросы, обоснования неверные, либо дан верный ответ без его

обоснования, сделаны грубые ошибки, отсутствуют знания нормативно-правовых документов по информационной безопасности.

Перечень вопросов к экзамену (3 семестр)

1. Особенности языка Python.
2. Структура программы в языке Python.
3. Понятие переменной.
4. Типы данных.
5. Инициализация переменных.
6. Основные типы переменных.
7. Понятие константы. Объявление. Инициализация.
8. Простейшие арифметические операции.
9. Принципы структурного программирования.
10. Оператор множественного ветвления switch.
11. Условный оператор if-else.
12. Вложенный оператор if.
13. Операторы цикла.
14. Одномерные массивы: объявление, инициализация, доступ к элементам.
15. Двумерные массивы: объявление, инициализация, доступ к элементам.
16. Доступ к элементам одномерного массива.
17. Доступ к элементам двумерного массива.
18. Стандартный класс string. Основные понятия.
19. Основные операции класса string.
20. Описание массива. Объявление и инициализация массива.
21. Описание строки. Объявление и инициализация строки.
22. Массив. Прямой доступ к элементам массива.
23. Стандартные методы обработки строк.
24. Принципы модульного программирования.
25. Понятие функции. Стандартные функции.
26. Описание функции.
27. Инструкция def.
28. Вызов функции.
29. Передача параметров в функцию.
30. Передача параметров по адресу.
31. Возврат значений из функций.
32. Анонимные функции.
33. Инструкция lambda.
34. Запись данных в файл.

Перечень вопросов к экзамену (4 семестр)

1. Принципы объектно-ориентированного программирования. Инкапсуляция.
2. Принципы объектно-ориентированного программирования. Полиморфизм.
3. Принципы объектно-ориентированного программирования. Наследование.
4. Классы. Объявление открытых переменных.
5. Классы. Инициализация открытых переменных.
6. Классы. Объявление закрытых переменных.
7. Классы. Инициализация закрытых переменных.
8. Спецификаторы доступа.

9. Функции-члены класса.
10. Связывание функций.
11. Встроенные функции.
12. Конструкторы. Основные понятия.
13. Конструкторы по умолчанию.
14. Конструкторы с параметрами.
15. Область действия класса. Управление доступом к членам класса.
16. Отделение интерфейса от реализации. Понятие интерфейса.
17. Понятие конструктора. Использование конструктора с аргументами по умолчанию. Конструктор по умолчанию.
18. Понятие деструктора. Когда вызываются конструкторы и деструкторы.
19. Константные объекты и функции-члены.
20. Дружественные функции и дружественные классы.
21. Использование указателя `this`.
22. Операции `new` и `delete`.
23. Статические члены класса.
18. Перегрузка функций.
19. Шаблоны функций.
20. Понятие перегрузки операций.
21. Функции-операции как члены класса и как дружественные функции.
22. Перегрузка операций ввода-вывода в поток, бинарных и унарных операций.
23. Перегрузка `++` и `--`.
24. Понятие наследования. Механизм ограничения доступа при наследовании.
25. Приведение типа указателя базового класса к указателю производного класса и наоборот.
26. Переопределение членов базового класса в производном классе.
27. Конструкторы и деструкторы в производных классах.
28. Неявное преобразование объектов производных классов в объекты базового класса.
29. Множественное наследование.
30. Понятие виртуальной функции.
31. Абстрактные и конкретные базовые классы.
32. Статическое и динамическое связывание.
33. Виртуальные деструкторы.
34. Виртуальные конструкторы.
35. Шаблоны классов.
36. Шаблоны классов и нетиповые параметры.
37. Шаблоны и статические члены.
38. Понятие исключения. Когда должна использоваться обработка исключений.
39. Генерация исключений. Повторная генерация исключений.
40. Перехватывание исключений.
41. Спецификация исключений.
42. Обработка неожиданных исключений.
43. «Раскручивание» стека. Иерархия исключений стандартной библиотеки.
44. Конструкторы, деструкторы и обработка исключений.
45. Исключения и наследование.
46. Обработка неуспешного выполнения `new`.

Критерии оценки экзамена:

– оценка «отлично» выставляется обучающемуся, если студент продемонстрировал глубокие знания теоретического материала и умение их применять, обоснованно изложил свои мысли, сделал необходимые выводы;

– оценка «хорошо» выставляется обучающемуся, если студент продемонстрировал глубокие знания теоретического материала и умение их применять, обоснованно изложил свои мысли, сделал необходимые выводы, допущены некоторые неточности, имеется одна негрубая ошибка;

– оценка «удовлетворительно» выставляется обучающемуся, если студент ответил на вопросы преимущественно верно, имеются затруднения в формулировке выводов, имеются одна или две негрубые ошибки;

– оценка «неудовлетворительно» выставляется обучающемуся, если студент не дал ответы на поставленные вопросы, обоснования неверные, либо дан верный ответ без его обоснования, сделаны грубые ошибки, отсутствуют знания по основам делопроизводства.

Таблица 9 – Примеры оценочных средств с ключами правильных ответов

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
ОПК 7 – способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности				
1.	Задание закрытого типа	Перевод программ с языка высокого уровня на язык более низкого уровня обеспечивает программа -: а) паскаль б) ассемблер в) компилятор	в	2
2.		Символьный тип данных объявляется служебным словом: а) STRING б) WORD в) CHAR	в	2
3.		В операторе присваивания <code>summa := sqr(x)+3*a</code> переменными являются: а) <code>sqr</code> , <code>x</code> , <code>a</code> б) <code>a</code> , <code>x</code> , <code>summa</code> + в) <code>summa</code> , <code>sqr</code> , <code>x</code> , <code>a</code>	б	2
4.		Шаблон функции - это... а) определение функции, в которой типу обрабатываемых данных присвоено условное обозначение б) прототип функции, в котором вместо имен параметров указан условный тип в) определение функции, в котором указаны возможные варианты типов обрабатываемых параметров г) определение функции, в котором в прототипе указан условный тип, а в определении	а	2

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
		указаны варианты типов обрабатываемых параметров		
5.		Что называется деструктором? а) метод, который уничтожает объект б) метод, который удаляет объект в) метод, который освобождает память, занимаемую объектом г) системная функция, которая освобождает память, занимаемую объектом	в	2
6.	Задание открытого типа	Средство, позволяющее использовать одно имя для обозначения действий, общих для родственных классов	Полиморфизм	2
7.		Ключевые элементы фон-неймановской архитектуры	Ключевыми элементами фон-неймановской архитектуры являются: память — оперативное запоминающее устройство (ОЗУ); центральный процессор (ЦП); арифметико-логическое устройство (АЛУ), способное выполнять базовые операции преобразования данных; устройство управления (УУ); счетчик команд (СК); устройства ввода-вывода.	2
8.		Основные операции над кортежами	К основным операциям над кортежами относятся: взятие элемента по его порядковому номеру; проверка на вхождение элемента; вырезка подкортежа; конкатенация; подсчет количества элементов	2
9.		Базовые операции над множествами	Базовые операции над множествами: проверка вхождения элемента в множество; теоретико-	2

№ п/п	Тип задания	Формулировка задания	Правильный ответ	Время выполнения (в минутах)
			множественные операции — объединение, пересечение, вычитание; сравнение множеств на эквивалентность; проверка вхождения подмножества в множество; удаление и/или добавление элемента.	
10.		Основные операции над файлами	Основные операции над файлами: открытие в различных режимах — на чтение, запись, добавление и пр. заккрытие после использования (обычно для фиксации изменений); чтение из файла — или порции байт, или структурированное; запись в файл — или порции байт, или структурированное; копирование файлов; переименование файла на диске; удаление файла.	2

Полный комплект оценочных материалов по дисциплине (модулю) (фонд оценочных средств) хранится в электронном виде на кафедре, утверждающей рабочую программу дисциплины (модуля), и в Центре мониторинга и аудита качества обучения.

7.4. Методические материалы, определяющие процедуры оценивания результатов обучения по дисциплине (модулю)

Методические рекомендации по выполнению лабораторных и контрольных работ, проведению экзамена

Отчет по лабораторной работе

Отчет по лабораторной работе представляется в электронном виде. Защита отчета проходит в форме доклада студента по выполненной работе и ответов на вопросы преподавателя. В случае, если оформление отчета и поведение студента во время защиты соответствуют указанным требованиям, студент получает максимальное количество баллов.

Основаниями для снижения количества баллов в диапазоне от max до min являются:

- отсутствие списка использованной литературы,
- небрежное выполнение,

- отсутствие выводов.

Отчет не может быть принят и подлежит доработке в случае:

- отсутствия необходимых разделов,
- отсутствия необходимого графического материала,
- неверных результатов расчета.

В отчете по выполненной лабораторной работе должны быть указаны:

- тема лабораторной работы,
- пакет документов в соответствии с темой лабораторной работы,
- использованная литература.

Контрольные работы

Контрольная работа состоит из 2-х заданий.

Основаниями для снижения оценки за задание являются:

- ошибки в объяснениях и комментариях при верно выполненном задании;
- неполный ответ для теоретических заданий;
- небрежное выполнение;
- многократное переписывание контрольной работы.

Задание не может быть засчитано, если:

- даны два неверных ответа на теоретические вопросы.

Экзамен

Экзамен заключается в письменном ответе на 2 теоретических вопроса и устном собеседовании по каждому теоретическому вопросу.

Основаниями для снижения оценки за теоретический вопрос являются:

- небрежное выполнение;
- неполный ответ;
- наличие мелких неточностей или незначительных искажений фактов;
- неточные объяснения при собеседовании;
- отсутствие ответов на заданные при собеседовании вопросы.

Оценивание студентов на экзамене осуществляется в соответствии с требованиями и критериями 100-балльной шкалы. Учитываются как результаты текущего контроля, так и знания, навыки и умения, непосредственно показанные студентами в ходе экзамена.

В соответствии с балльно-рейтинговой системой БАРС по дисциплине отводится 100 баллов (40 баллов на текущие формы контроля, 10 баллов на бонусы и 50 баллов отводится на экзамен),

Оценивание студентов на экзамене осуществляется в соответствии с требованиями и критериями 100-балльной шкалы. Учитываются как результаты текущего контроля, так и знания, навыки и умения, непосредственно показанные студентами в ходе экзамена.

Критерии оценок на экзамене:

40-50 баллов – студент глубоко понимает пройденный материал, отвечает четко и всесторонне, умеет оценивать факты, самостоятельно рассуждает, отличается способностью обосновать выводы и разъяснять их в логической последовательности.

35-39 баллов – студент глубоко понимает пройденный материал, отвечает четко и всесторонне, умеет оценивать факты, самостоятельно рассуждает, отличается способностью обосновать выводы и разъяснять их в логической последовательности, но допускает отдельные неточности.

25-34 балла – студент глубоко понимает пройденный материал, отвечает четко и всесторонне, умеет оценивать факты, самостоятельно рассуждает, отличается способностью обосновать выводы и разъяснять их в логической последовательности, но допускает некоторые ошибки общего характера.

20-24 балла – студент хорошо понимает пройденный материал, но не может теоретически обосновать некоторые выводы.

15-19 баллов – студент отвечает в основном правильно, но чувствуется механическое заучивание материала.

11-14 баллов – в ответе студента имеются существенные недостатки, материал охвачен «половинчато», в рассуждениях допускаются ошибки.

10 баллов – ответ студента правилен лишь частично, при разъяснении материала допускаются серьезные ошибки.

6-9 баллов – студент имеет общее представление о теме, но не умеет логически обосновать свои мысли.

1-5 баллов – студент имеет лишь частичное представление о теме. 0 баллов – нет ответа.

Таблица 10 – Технологическая карта рейтинговых баллов по дисциплине (модулю) в 2 семестре

№ п/п	Контролируемые мероприятия	Количество мероприятий / баллы	Максимальное количество баллов	Срок представления
Основной блок				
1.	<i>Выполнение лабораторной работы</i>	11/2	22	По расписанию
2.	<i>Выполнение контрольной работы</i>	3/5	15	
3.	<i>Проект</i>	1/3	3	
Всего			40	-
Блок бонусов				
4.	<i>Посещение занятий без пропусков</i>	1	3	
5.	<i>Своевременное выполнение всех заданий</i>	1	3	
6.	<i>Активность студента на занятии</i>	1	4	
Всего			10	-
Дополнительный блок				
7.	<i>Экзамен</i>		50	
Всего			50	-
ИТОГО			100	-

Таблица 10 – Технологическая карта рейтинговых баллов по дисциплине (модулю) в 3 семестре

№ п/п	Контролируемые мероприятия	Количество мероприятий / баллы	Максимальное количество баллов	Срок представления
Основной блок				
1.	<i>Выполнение лабораторной работы</i>	8/2	16	По расписанию
2.	<i>Выполнение контрольной работы</i>	3/5	15	
3.	<i>Проект</i>	1/9	9	
Всего			40	-
Блок бонусов				
4.	<i>Посещение занятий без пропусков</i>	1	3	

№ п/п	Контролируемые мероприятия	Количество мероприятий / баллы	Максимальное количество баллов	Срок представления
5.	<i>Своевременное выполнение всех заданий</i>	1	3	
6.	<i>Активность студента на занятии</i>	1	4	
Всего			10	-
Дополнительный блок				
7.	<i>Экзамен</i>		50	
Всего			50	-
ИТОГО			100	-

Таблица 11 – Система штрафов (для одного занятия)

Показатель	Балл
<i>Опоздание на занятие</i>	- 1
<i>Нарушение учебной дисциплины</i>	- 1
<i>Неготовность к занятию</i>	- 2
<i>Пропуск занятия без уважительной причины</i>	- 2

Таблица 12 – Шкала перевода рейтинговых баллов в итоговую оценку за семестр по дисциплине (модулю)

Сумма баллов	Оценка по 4-балльной шкале	
90–100	5 (отлично)	
85–89	4 (хорошо)	
75–84		
70–74		
65–69	3 (удовлетворительно)	
60–64		
Ниже 60	2 (неудовлетворительно)	

При реализации дисциплины (модуля) в зависимости от уровня подготовленности обучающихся могут быть использованы иные формы, методы контроля и оценочные средства, исходя из конкретной ситуации.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8.1. Основная литература:

1. Барышникова М.Ю., Основы программирования на C/C++. Ч. 2 [Электронный ресурс] : Учеб. пособие / Барышникова М.Ю., Силантьева А.В. - М. : Издательство МГТУ им. Н. Э. Баумана, 2007. - 70 с. - ISBN 978-5-7038-2939-4 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785703829394.html>

2. Уйманова Н.А., Основы объектно-ориентированного программирования [Электронный ресурс] / Уйманова Н.А. - Оренбург: ОГУ, 2017. - 768 с. - ISBN 978-5-7410-1993-1 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785741019931.html>

8.2. Дополнительная литература:

1. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Д. Приемы объектно-ориентированного проектирования. Паттерны проектирования: Пер. с англ. - М.: ДМК Пресс, 2011- 368 с.: ил. <http://www.studentlibrary.ru>
2. Батоврин В.К. Системная и программная инженерия. Словарь-справочник: учеб. пособие для вузов. – М.: ДМК Пресс, 2010. - 280 с.: ил. <http://www.studentlibrary.ru>
3. Алгоритмы и структуры данных. Новая версия для Оберона/ Алгоритмы и структуры данных. Новая версия для Оберона + CD / Пер. с англ. Ткачев Ф. В. - М.: ДМК Пресс, 2010. - 272 с.: ил <http://www.studentlibrary.ru>

8.3. Интернет-ресурсы, необходимые для освоения дисциплины (модуля)

1. **Электронно-библиотечная система (ЭБС) ООО «Политехресурс» «Консультант студента».** Многопрофильный образовательный ресурс «Консультант студента» является электронной библиотечной системой, предоставляющей доступ через сеть Интернет к учебной литературе и дополнительным материалам, приобретенным на основании прямых договоров с правообладателями. Каталог в настоящее время содержит около 15000 наименований. www.studentlibrary.ru.

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Учебные аудитории, библиотеки АГУ, компьютерные классы, мультимедийные аудитории.

Рабочая программа дисциплины (модуля) при необходимости может быть адаптирована для обучения (в том числе с применением дистанционных образовательных технологий) лиц с ограниченными возможностями здоровья, инвалидов. Для этого требуется заявление обучающихся, являющихся лицами с ограниченными возможностями здоровья, инвалидами, или их законных представителей и рекомендации психолого-медико-педагогической комиссии. Для инвалидов содержание рабочей программы дисциплины (модуля) может определяться также в соответствии с индивидуальной программой реабилитации инвалида (при наличии).