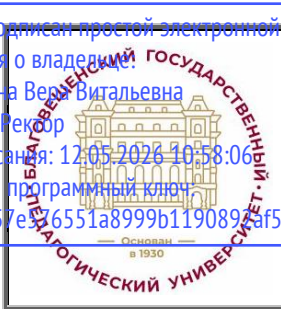



Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Щёкина Вера Витальевна
Должность: Ректор
Дата подписания: 12.05.2026 10:58:06
Уникальный программный ключ:
a2232a55157e376551a8999b1190897af5398942642d556b0c375a454e57789

	МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
	федеральное государственное бюджетное образовательное учреждение высшего образования «Благовещенский государственный педагогический университет»
	ОСНОВНАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА Рабочая программа дисциплины

УТВЕРЖДАЮ
Декан факультета физико-математического
образования и технологии
ФГБОУ ВО «БГПУ»

Н.В. Слесаренко
«03» сентября 2024 г.

**Рабочая программа дисциплины
ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ**

**Направление подготовки
09.03.02 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ**

**Профиль
«ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ»**

**Уровень высшего образования
БАКАЛАВРИАТ**

**Принята на заседании кафедры
информатики и МПИ
(протокол № 8 от «29» мая 2024 г.)**

Благовещенск 2024

СОДЕРЖАНИЕ

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
2 УЧЕБНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ	4
3 СОДЕРЖАНИЕ ТЕМ (РАЗДЕЛОВ)	6
4 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ (УКАЗАНИЯ) ДЛЯ СТУДЕНТОВ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ	6
5 ПРАКТИКУМ ПО ДИСЦИПЛИНЕ	8
6 ДИДАКТИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ КОНТРОЛЯ (САМОКОНТРОЛЯ) УСВОЕННОГО МАТЕРИАЛА.....	13
7 ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ	23
В ПРОЦЕССЕ ОБУЧЕНИЯ	23
8 ОСОБЕННОСТИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ ИНВАЛИДАМИ И ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ	23
9 СПИСОК ЛИТЕРАТУРЫ И ИНФОРМАЦИОННЫХ РЕСУРСОВ	23
10 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА	24
11 ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ	26

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1 Цель дисциплины: выработка практических навыков алгоритмизации классических алгоритмов и их реализации, приобретение навыков по выбору оптимального инструментария из арсенала С++ для решения конкретных вычислительных задач различных областей деятельности, способствовать дальнейшему развитию алгоритмического мышления объектно-ориентированного подхода.

1.2 Место дисциплины в структуре ООП: Дисциплина «Практикум по программированию» относится к дисциплинам части, формируемой участниками образовательных отношений, блока Б1 (Б1.В.03).

Для освоения дисциплины «Практикум по программированию» используются знания, умения и виды деятельности, сформированные в процессе изучения дисциплины «Технология программирования». Дисциплина «Практикум по программированию» в профессиональной подготовке выпускника является базовой для изучения других программистских дисциплин, изучение которой позволит студентам выработать современный подход к качеству и содержанию компьютерных программ. Дисциплина «Практикум по программированию» обеспечивает раскрытие общего круга вопросов разработки программного обеспечения, алгоритмов, организации научных вычислений и моделирования. В ходе изучения дисциплины разбираются классические вычислительные алгоритмы, алгоритмы обработки нечисловых данных, студенты получают представление об эффективности алгоритмов и представления об анализе их эффективности. Выработанные навыки алгоритмизации и программирования необходимы для изучения таких дисциплин как «Современные технологии программирования» и др.

1.3 Дисциплина направлена на формирование следующих компетенций: ОПК-6:

- **ОПК-6.** Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий, **индикаторами** достижения которой является:

- ИД-1опк-б-знать: основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий.
- ИД-2опк-б-уметь: применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.
- ИД-3опк-б-иметь навыки: программирования, отладки и тестирования прототипов программно-технических комплексов задач.

1.4 Перечень планируемых результатов обучения. В результате изучения дисциплины студент должен

знать:

- принципы автономной отладки и тестирования простых программ;
- принципы, базовые концепции технологий программирования;
- основные этапы и принципы создания программного продукта;

уметь:

- профессионально грамотно сформулировать задачу программирования;
- разрабатывать алгоритмы решения;
- реализовать задачу обработки данных в предметной области в заданной языковой среде, типа CodeBlocks или Dev-C++;
- использовать языки и системы программирования для решения профессиональных задач;

- выполнить необходимое тестирование, отладку или верификацию программы;
 - применять информационные технологии при проектировании информационных систем;
 - анализировать и улучшать производительность создаваемых программ;
- владеть:**
- языками процедурного и объектно-ориентированного программирования;
 - навыками одной из технологий программирования.

1.5 Общая трудоемкость дисциплины «Практикум по программированию» составляет 2 зачетные единицы (далее – ЗЕ) (72 часа).

Программа предусматривает изучение материала на лекциях и практических занятиях. Предусмотрена самостоятельная работа студентов по темам и разделам. Проверка знаний осуществляется фронтально, индивидуально.

1.6 Объем дисциплины и виды учебной деятельности

Объем дисциплины и виды учебной деятельности (очная форма обучения)

Вид учебной работы	Всего часов	Семестр 3
Общая трудоемкость	72	72
Аудиторные занятия	36	36
Лекции	-	-
Лабораторные работы	36	36
Самостоятельная работа	36	36
Вид итогового контроля	-	зачёт

Объем дисциплины и виды учебной деятельности (заочная форма обучения)

Вид учебной работы	Всего часов	Семестр 4
Общая трудоемкость	72	72
Аудиторные занятия	10	10
Лекции	-	-
Лабораторные работы	10	10
Самостоятельная работа	58	58
Вид итогового контроля	4	зачёт

2 УЧЕБНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

2.1 Очная форма обучения

Учебно-тематический план

№	Наименование тем (разделов)	Всего часов	Аудиторные занятия		Самостоятельная работа
			Лекции	Лабораторные работы	
1.	Принципы объектно-ориентированного подхода к разработке программ. Определение класса: поля, методы, объекты.	20		10	10
2.	Инкапсуляция и спецификация правил доступности элементов классов. Наследование.	8		4	4

3.	Виртуальные функции.	12		6	6
4.	Перегрузка функций и операций.	8		4	4
5.	Шаблоны.	12		6	6
6.	Контейнеры.	12		6	6
Зачёт					
ИТОГО		72		36	36

Интерактивное обучение по дисциплине

№	Наименование тем (разделов)	Вид занятия	Форма интерактивного занятия	Кол-во часов
1.	Виртуальные функции. Наследование.	Лб.	Обратная связь	2
2.	Перегрузка функций и операций.	Лб.	Работа в малых группах	2
3.	Шаблоны.	Лб.	Работа в малых группах	2
4.	Контейнеры.	Лб.	Обратная связь	2
ИТОГО				8

4.2 Заочная форма обучения

Учебно-тематический план

№	Наименование тем (разделов)	Всего часов	Аудиторные занятия		Самостоятельная работа
			Лекции	Лабораторные работы	
1.	Принципы объектно-ориентированного подхода к разработке программ. Определение класса: поля, методы, объекты.	18		4	14
2.	Инкапсуляция и спецификация правил доступности элементов классов. Наследование.	8		2	6
3.	Виртуальные функции.	10		2	8
4.	Перегрузка функций и операций.	8		0	8
5.	Шаблоны.	12		2	10
6.	Контейнеры.	12			12
Зачёт		4			
ИТОГО		72		10	58

Интерактивное обучение по дисциплине

№	Наименование тем (разделов)	Вид занятия	Форма интерактивного	Кол-во
---	-----------------------------	-------------	----------------------	--------

			занятия	часов
1.	Виртуальные функции. Наследование.	Лб.	Обратная связь	2
2.	Перегрузка функций и операций.	Лб.	Работа в малых группах	2
	ИТОГО			4

3 СОДЕРЖАНИЕ ТЕМ (РАЗДЕЛОВ)

1. ПРИНЦИПЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА К РАЗРАБОТКЕ ПРОГРАММ. ОПРЕДЕЛЕНИЕ КЛАССА: ПОЛЯ, МЕТОДЫ, ОБЪЕКТЫ.

Изучение и постановка классических задач обработки данных. Основные свойства и понятия объектно-ориентированного программирования.

2. ИНКАПСУЛЯЦИЯ И СПЕЦИФИКАЦИЯ ПРАВИЛ ДОСТУПНОСТИ ЭЛЕМЕНТОВ КЛАССОВ. НАСЛЕДОВАНИЕ

Поиск и изучение в литературе описания структур данных: очередей, стека, кучи, деревьев. Изучение примеров их реализаций. Линейный односвязный список. Проход по списку, поиск элемента в списке, включение и исключение элементов.

3. ВИРТУАЛЬНЫЕ ФУНКЦИИ

Поиск и изучение в литературе описания алгоритмов: реализация алгоритмов сортировки, поиска, обхода деревьев. Сравнение алгоритмов. Изучение вопроса эффективности алгоритма.

4. ПЕРЕГРУЗКА ФУНКЦИЙ И ОПЕРАЦИЙ

Необходимость перегрузки функций и операций. Оформление перегрузки операций, правила перегрузки, выбор типа функции-операции.

5. ПРАКТИКА ПО ИСПОЛЬЗОВАНИЮ СТОРОННИХ БИБЛИОТЕК. ШАБЛОНЫ

Изучение API сторонних библиотек. Понятия шаблона.

6. КОНТЕЙНЕРЫ

Топологическая специфика методологий. Модульное программирование. Независимость программирования модулей. Блочная вложенность и локализация. Правила видимости. Классы памяти: auto, register, static, extern. Внутренняя и внешняя документация программ. Структура программы: модули, объекты (классы), функции (построение и использование).

4 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ (УКАЗАНИЯ) ДЛЯ СТУДЕНТОВ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ

4.1 Общие методические рекомендации

Излагаемая дисциплина основывается на фундаментальных понятиях информатики и программирования, в то же время имеет четкую прикладную направленность. При разработке рабочей программы дисциплины предусмотрено, что определенные вопросы изучаются студентами самостоятельно.

Практикум по дисциплине представлен учебно-методическим материалом по подготовке к лабораторным занятиям, размещенным в системе электронного обеспечения БГПУ. Дидактические материалы для контроля (самоконтроля) усвоения учебного

материала содержат вопросы допуска к лабораторным работам, примерные вопросы зачета, а так же индивидуальные. Раздел программы «Список литературы и информационных ресурсов» – позволяет использовать материалы не только для подготовки к аудиторным занятиям, но и для организации самостоятельной работы, а также для расширения собственных представлений по отдельным разделам изучаемой дисциплины.

Практикум по дисциплине включает:

- тематику и план лабораторных занятий;
- вопросы допуска для подготовки к лабораторным работам;
- список литературы и информационных ресурсов, необходимых для целенаправленной подготовки студентов к каждому занятию.

Материалы практикума направлены на глубокое изучение важнейших проблем программирования, предусматривают сочетание аудиторных и внеаудиторных форм организации учебного процесса.

Основное предназначение дидактических материалов – помочь студентам организовать самостоятельную подготовку по дисциплине, провести самоконтроль умений и знаний, получить чёткое представление о предстоящих формах контроля.

4.2 Методические рекомендации по подготовке к лабораторным занятиям

Целью лабораторных занятий является выработка у студентов навыков практической работы на компьютере, выработка навыков алгоритмизации, умения составлять программный код на одном из алгоритмических языков. Целью лабораторных работ является закрепление теоретического материала, изложенного на предыдущем этапе в дисциплине «Программирование и выработка умения решать задачи.

Для проведения лабораторных работ используются компьютеры, оснащенные ОС Windows 2000 и выше, интегрированной средой DevC++ или CodeBlocks. Возможно использование проектора или интерактивной доски.

По каждой теме предлагаются примеры задач для выполнения на лабораторных занятиях, в конце каждой лабораторной предлагаются задания для самостоятельного решения, а также варианты индивидуальных заданий.

При выполнении лабораторной работы недостаточно набрать программу без ошибок, необходимо тщательно проанализировать полученные результаты для различных исходных данных, а также выполнить задания, предлагаемые к некоторым программам.

Варианты индивидуальных заданий назначаются преподавателем. Индивидуальные задания выполняются студентом как на занятиях, так и дома в рамках самостоятельной работы студента и сдаются преподавателю в виде отчетов. Необходимо продумать всевозможные исходные данные и продемонстрировать работу программы преподавателю в различных ситуациях. В случае неверно введенных данных должно быть выдано соответствующее сообщение на экран. Если для введенного значения нет данных, удовлетворяющих заданному условию, также вывести об этом сообщение на экран.

Выполнение индивидуальных заданий оформлять в виде отчета. Содержание отчета:

- тема выполняемой индивидуальной работы;
- тексты программ;
- результаты отладки (ошибки, их причины и способы устранения);
- результаты работы программы для различных исходных данных.

Условием сдачи зачета является выполнение лабораторных работ, отчет по всем индивидуальным заданиям.

Лабораторные работы в компьютерных классах служат для выполнения студентами учебных задач с целью выработки и закрепления практических навыков программирования.

В случае появления каких-либо вопросов следует обращаться к преподавателю в часы его консультаций. Критерием качества усвоения знаний могут служить аттестацион-

ные оценки по дисциплине и текущие оценки, выставяемые преподавателем в течение семестра.

При решении практических задач используются интерактивные методы обучения, позволяющие интенсифицировать процесс понимания, усвоения и творческого применения студентами полученных знаний, повысить мотивацию и вовлеченность их в решение обсуждаемых проблем, что дает эмоциональный толчок к последующей поисковой активности обучающихся, побуждает их к конкретным действиям, процесс обучения становится более осмысленным.

4.3 Методические указания к самостоятельной работе студентов

Для успешного усвоения дисциплины необходима правильная организация самостоятельной работы студентов. Эта работа должна содержать:

- регулярную (еженедельную) подготовку к лабораторным занятиям, в том числе изучение описания лабораторных работ;
- регулярное (еженедельное) решение индивидуальных задач, задаваемых преподавателем.

4.5 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине:

1. Фонд оценочных средств.
2. Вопросы к экзамену и зачёту.
3. Список литературы и информационных ресурсов.

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

№	Наименование раздела (темы)	Формы/виды самостоятельной работы	Количество часов, в соответствии с учебно-тематическим планом
1.	Принципы объектно-ориентированного подхода к разработке программ. Определение класса: поля, методы, объекты.	Чтение рекомендованной литературы	10
2.	Инкапсуляция и спецификация правил доступности элементов классов. Наследование.	Чтение рекомендованной литературы	4
3.	Перегрузка функций и операций.	Чтение рекомендованной литературы	4
4.	Виртуальные функции.	Чтение рекомендованной литературы	6
5.	Шаблоны.	Чтение рекомендованной литературы	6
6.	Контейнеры.	Чтение рекомендованной литературы	6
	ИТОГО		36

5 ПРАКТИКУМ ПО ДИСЦИПЛИНЕ

Принципы объектно-ориентированного подхода к разработке программ. Определение класса: поля, методы, объекты.

Лабораторная работа № 1. Пример класса Дробь. (4 часа)

Лабораторная работа № 2. Описание класса. Конструкторы и деструктор. (6 часов)

Инкапсуляция и спецификация правил доступности элементов классов. Наследование.

Лабораторная работа № 3. Наследование. (4 часа)

Перегрузка функций и операций.

Лабораторная работа № 4. Полиморфизм: перегрузка операций. (4 часа)

Виртуальные функции.

Лабораторная работа № 5. Полиморфизм: виртуальные функции. (6 часов)

Шаблоны

Лабораторная работа № 6. Встроенные классы библиотеки ifstream.h (6 часов)

Контейнеры

Лабораторная работа № 7 Контейнерные классы типа Вектор. (4 часов)

Лабораторная работа № 8 Контейнерные классы типа Стек. (2 часов)

Итого: 36 ч.

Содержание

Лабораторная работа №1 Пример класса Дробь. (4 часа)

Тема: *Принципы объектно-ориентированного подхода к разработке программ. Определение класса: поля, методы, объекты.*

Задание. Общее групповое задание: разобрать и отладить пример класса Дробь, применить класс Дробь к решению системы уравнений с двумя неизвестными.

Лабораторная работа №2 Описание класса. Конструкторы и деструктор. (4 часа)

Тема: *Принципы объектно-ориентированного подхода к разработке программ. Определение класса: поля, методы, объекты.*

Задание. В соответствии с вариантом составить описание класса. Класс должен обязательно содержать не менее двух конструкторов – по умолчанию и с параметрами, деструктор. Объявить объекты класса и продемонстрировать работу методов.

Варианты

1. Организовать класс матрица, содержащий методы вывода матрицы в общепринятом виде, нахождения транспонированной матрицы и определителя матрицы.
2. Организовать класс дробь, содержащий методы вывода дроби в общепринятом виде и функцию выделения целой части.
3. Организовать класс дробь, содержащий методы вывода дроби в общепринятом виде и функцию приведения дроби к несократимому виду.

4. Описать класс вектор на плоскости, содержащий координаты его начала и конца и методы его изображения на экране и нахождения длины.
5. Описать класс многочлен, с полями степень, аргумент и коэффициенты. Создать метод вычисления значения многочлена от аргумента и вывода многочлена в общем виде на экран.
6. Описать класс множество с полями элементы множества и количество элементов, позволяющий добавлять и удалять элементы из множества, вывод всей элементов множества на экран.
7. Создать класс комплексное число с методами нахождения модуля и аргумента комплексного числа и вывода числа на экран в общепринятом виде.
8. Организовать класс вектор, содержащий его координаты в трехмерном пространстве и методы нахождения длины вектора, вывода на экран координат.

Лабораторная работа № 3 Наследование. (4 часа)

Тема Инкапсуляция и спецификация правил доступности элементов классов. Наследование.

Задание. Изменить программу предыдущего задания, организовав на основе существующего класса производный класс. Производный класс также должен содержать не менее двух конструкторов.

Исходная программа должна содержаться в двух файлах. В первом описание классов, во втором реализация, т.е. работа с объектами. Продемонстрировать работу всех методов базового и производного классов.

Варианты

1. Организовать производный класс, дополнительно содержащий функции нахождения произведения и сложения матриц.
2. Организовать производный класс, содержащий функции вычисления сложения и вычитания дробей.
3. Организовать производный класс, содержащий функции вычисления деления и умножения дробей, функцию приведения дроби к несократимому виду.
4. Описать производный класс, дополнительно содержащий функцию умножения вектора на число и сложения двух векторов.
5. Описать производный класс, позволяющий умножать многочлен на число и складывать многочлены.
6. Описать производный класс, дополнительно содержащий функции объединения и пересечения множеств.
7. Описать производный класс, дополнительно содержащий функции сложения и вычитания комплексных чисел.
8. Организовать производный класс, наследующий все свойства базового и дополнительный методы нахождения скалярного произведения и сложения векторов.

Лабораторная работа № 4 Полиморфизм: перегрузка операций. (4 часа)

Тема Перегрузка функций и операций.

Варианты

1. Изменить программу предыдущей лабораторной работы, используя для сложения и умножения матриц перегрузку соответствующих операций.
2. Изменить программу предыдущей лабораторной работы, используя для сложения и вычитания дробей перегрузку соответствующих операций.

3. Изменить программу предыдущей лабораторной работы, используя для деления и умножения матриц перегрузку соответствующих операций.
4. Изменить программу предыдущей лабораторной работы, используя для сложения и умножения векторов перегрузку соответствующих операций.
5. Изменить программу предыдущей лабораторной работы, используя для сложения и умножения многочленов перегрузку соответствующих операций.
6. Изменить программу предыдущей лабораторной работы, используя для сложения и вычитания множеств перегрузку соответствующих операций.
7. Изменить программу предыдущей лабораторной работы, используя для сложения и вычитания комплексных чисел перегрузку соответствующих операций.
8. Изменить программу предыдущей лабораторной работы, используя перегрузку операции умножения, для метода нахождения скалярного произведения.

Лабораторная работа № 5 *Полиморфизм: виртуальные функции* (6 часов)

Тема Виртуальные функции.

Варианты

1. Описать абстрактный класс животное. Класс должен содержать характеристики животных: название, вид, местообитание, функцию вывода всех данных на экран. На его основе реализовать классы Млекопитающее, Рыба, Птица. Отдельными характеристиками классов являются: для млекопитающих – травоядное, хищник или всеядное; для рыб – морская или пресноводная; для птиц – дикая, домашняя, если дикая перелетная, или нет.
2. Создать абстрактный класс средство передвижения. На его основе реализовать классы самолет, машина, корабль. Все классы должны хранить параметры средств передвижения: скорость, расход топлива, наименование производителя, год выпуска, метод вывода на экран всех данных, определения срока службы. Индивидуально для самолета указать высоту и максимальную дальность полета, для машины – объем двигателя, для самолета и корабля – количество посадочных мест, для корабля – водоизмещение.
3. Создать абстрактный класс правильный многоугольник. На его основе создать классы треугольник, квадрат, пятиугольник. Предусмотреть методы создания объектов, вычисления их периметра, площади, величины угла.
4. Описать абстрактный класс фигура на плоскости. На его базе создать классы круг, треугольник, прямоугольник. Предусмотреть методы создания объектов, вычисления площади фигур, периметра для треугольника и прямоугольника, длины окружности – для круга.
5. Создать абстрактный класс правильный многоугольник. На его основе создать классы треугольник, квадрат, восьмиугольник. Предусмотреть методы создания объектов, вычисления их периметра, площади, величины угла.
6. Создать абстрактный класс правильный многогранник с полями длина ребра и число ребер. На его основе создать классы тетраэдр, куб, октаэдр (восьмигранник). Предусмотреть методы создания объектов, вычисления их площади поверхности и объема.
7. Создать абстрактный класс вектор. На его основе создать классы вектор на плоскости, в трехмерном пространстве, в пятимерном пространстве. Предусмотреть методы создания объектов, вычисления их длины, вывода на экран их координат.
8. Создать абстрактный класс человек. На его основе создать классы школьник, студент, преподаватель.

Лабораторная работа № 6 *Встроенные классы библиотеки *ifstream.h (6 часов)**

Тема *Шаблоны*

Варианты

1. Создать шаблон класса «бинарное дерево». Использовать его для сортировки целых чисел и строк, задаваемых с клавиатуры.
2. Создать шаблон для работы с комплексными числами. Использовать его, задавая мнимую и вещественную части числами типа double, int, float.
3. Создать Шаблон класса «очередь». Использовать его для различных типов элементов. Продемонстрировать в программе работу всех методов класса и для различных типов параметров. Программа должна содержать меню, пункты которого соответствуют названиям методов.
4. Создать Шаблон класса «однонаправленный линейный список». Использовать его для различных типов элементов. Продемонстрировать в программе работу всех методов класса и для различных типов параметров. Программа должна содержать меню, пункты которого соответствуют названиям методов.

Лабораторная работа № 7 *Контейнерные классы типа Вектор (4 часа)*

Тема *Контейнеры*

Задание: Выполнить задание с использованием контейнерных классов стандартной библиотеки C++

Варианты

1. Определить сумму положительных элементов, расположенных после минимального элемента данного массива.
2. Определить среднее арифметическое отрицательных элементов, расположенных после минимального элемента данного массива.
3. Определить произведение положительных элементов, расположенных до максимального элемента этого массива.
4. Определить сумму положительных элементов, расположенных между минимальным и максимальным элементами данного массива.
5. Определить среднее геометрическое отрицательных элементов, расположенных между минимальным и максимальным элементами данного массива.
6. Определить сумму минимального и максимального элементов данного массива.
7. Определить сумму положительных элементов, расположенных после минимального элемента данного массива.
8. Определить минимальный элемент среди положительных элементов данного массива.

Лабораторная работа № 8 *Контейнерные классы типа Стек (2 часа)*

Тема *Контейнеры*

Задание: Выполнить задание с использованием контейнерных классов стандартной библиотеки C++

Выполнить задание лабораторной работы № 6, используя контейнерный класс Стек

6 ДИДАКТИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ КОНТРОЛЯ (САМОКОНТРОЛЯ) УСВОЕННОГО МАТЕРИАЛА

6.1 Оценочные средства, показатели и критерии оценивания компетенций

Индекс компетенции	Оценочное средство	Показатели оценивания	Критерии оценивания сформированности компетенций
ОПК-6	Опрос	Низкий (неудовлетворительно)	Студент отвечает неправильно, нечетко и неубедительно, дает неверные формулировки, в ответе отсутствует какое-либо представление о вопросе
		Пороговый (удовлетворительно)	Студент отвечает неконкретно, слабо аргументировано и не убедительно, хотя и имеется какое-то представление о вопросе
		Базовый (хорошо)	Студент отвечает в целом правильно, но недостаточно полно, четко и убедительно, путается в понятиях
		Высокий (отлично)	Ставится, если продемонстрированы знание вопроса и самостоятельность мышления, ответ соответствует требованиям правильности, полноты и аргументированности.
ОПК-6	Лабораторная работа Оценивается рубрикой в СЭО БГПУ	Низкий (неудовлетворительно)	обнаруживает незнание большей части соответствующего вопроса, допускает ошибки в заданиях, не понимает их смысл, беспорядочно и неуверенно излагает теорию по теме лабораторной.
		Пороговый (удовлетворительно)	допускает неточности при выполнении заданий; путается в пояснительных комментариях к примерам; не умеет сделать выводы;
		Базовый (хорошо)	выполняет задания по лабораторной работе не в полном объеме; в пояснительных комментариях к примерам допускает неточности; делает выводы по аналогии к примерам самостоятельной работы;
		Высокий (отлично)	выполняет полный объем заданий по лабораторной работе; дает полные пояснительные комментарии к примерам; обнаруживает понимание при ответе на контрольные вопросы, может обосновать полученные результаты;

ОПК-6	Разноуровневые задачи и задания (домашние и индивидуальные задания)	Низкий (неудовлетворительно)	<p>Ответ студенту не зачитывается если:</p> <ul style="list-style-type: none"> • задание выполнено менее, чем на половину; • студент обнаруживает незнание большей части соответствующего материала, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно излагает материал.
		Пороговый (удовлетворительно)	<p>Задание выполнено более, чем на половину. Студент обнаруживает знание и понимание основных положений задания, но:</p> <ul style="list-style-type: none"> • излагает материал неполно и допускает неточности в определении понятий; • не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; • излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.
		Базовый (хорошо)	<p>Задание в основном выполнено. Ответы правильные, но:</p> <ul style="list-style-type: none"> • в ответе допущены малозначительные ошибки и недостаточно полно раскрыто содержание вопроса; • не приведены иллюстрирующие примеры, недостаточно чётко выражено обобщающее мнение студента; • допущено 1-2 недочета в последовательности и языковом оформлении излагаемого.
		Высокий (отлично)	<p>Задание выполнено в максимальном объеме. Ответы полные и правильные.</p> <ul style="list-style-type: none"> • студент полно излагает материал, дает правильное определение основных понятий; • обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры; • излагает материал последовательно и правильно с точки зрения норм литературного языка.
ОПК-6	Итоговый тест	Низкий (неудовлетворительно)	Количество правильных ответов на вопросы теста менее 60 %
		Пороговый (удовлетворительно)	<ul style="list-style-type: none"> • Количество правильных ответов на вопросы теста от 61-75 %
		Базовый (хорошо)	<ul style="list-style-type: none"> • Количество правильных ответов на вопросы теста от 76-84 %
		Высокий (отлично)	Количество правильных ответов на вопросы теста от 85-100 %

6.2 Промежуточная аттестация студентов по дисциплине

Промежуточная аттестация является проверкой всех знаний, навыков и умений студентов, приобретённых в процессе изучения дисциплины. Формой промежуточной аттестации по дисциплине является зачёт.

Для оценивания результатов освоения дисциплины применяется следующие критерии оценивания.

Критерии оценивания устного ответа на зачете

Оценка «зачтено» выставляется студенту, если:

- задания, размещенные в Электронной информационно-образовательной среде БГПУ выполнены на 60 и более процентов;
- имеет посещаемость лабораторных занятий не менее 60 процентов.

Оценка «не зачтено» выставляется студенту, если:

- задания, размещенные в Электронной информационно-образовательной среде БГПУ выполнены менее чем на 60 процентов;
- имеет посещаемость лабораторных занятий менее 60 процентов (исключение составляют студенты, пропустившие занятия по уважительной причине: болезни, участия в значимых для вуза мероприятиях, таких как участие в олимпиадах по профилю и т.п.).

Типовые контрольные задания или иные материалы, необходимые для оценки результатов освоения дисциплины

Пример тестовых вопросов

1. Выберите верное определение понятия «абстрактный класс»

- a) класс, не содержащий встроенные методы
- b) класс, содержащий встроенные методы
- c) класс, имеющий чистые виртуальные функции
- d) класс, не имеющий виртуальные функции

2. Что такое инкапсуляция?

- a) создание новых классов на базе уже имеющихся (базовых) классов, причем новые классы обладают всеми свойствами базовых классов и имеют новые, свойственные только им.
- b) создание сложных типов данных, включающих данные и методы для их обработки
- c) обозначение одним именем общего действия во всей иерархии типов
- d) определение функций, вызывающих саму себя.

3. Изучите программный код и сделайте выбор среди приведенных результатов

```
class A {
int x;
public:
    A ( int y=5) {x=y;}
    void PrintA (int _x) { cout<< "x = "<<x<<endl; }
};
int main
{A a;
a.PrintA();
}
```

- a) компилятор выдаст ошибку
 c) $x = 5$
- b) $x = 0$
 d) 5

4. Для фрагмента программы из предыдущего пункта какой спецификатор доступа поле x имеет?

- a) public b) private c) protected d) void

5. Что называю методом класса?

- a) элемент класса, содержащий данные
 b) элемент класса, содержащий функцию
 c) функцию, отвечающую за инициализацию данных в классе
 d) функцию, отвечающую за уничтожение данных в классе

6. Выберите верное утверждение о множественном наследовании

- a) создание новых классов на основе одного базового класса
 b) создание в одной программе нескольких новых классов на основе множества классов
 c) создание нового класса на основе нескольких базовых классов
 d) создание нескольких классов в одной программе

7. Какой принцип ООП означает обозначение одним именем общего действия во всей иерархии типов?

- a) инкапсуляция б) полиморфизм в) наследование г) рекурсия

8. Выберите правильный вариант результата работы программы

```
class A { int a;
public: A() {cout<< "construct A ";}
~A() {cout<< "destruct A ";}
};
class B : class A { int a;
public: B() {cout<< "construct B ";}
~B() {cout<< "destruct B ";}
};
int main ()
{B b;}
```

- a) ничего не выведется
 b) construct B destruct B
 c) construct A destruct A construct B destruct B
 d) construct A construct B destruct B destruct A
 e) construct B construct A destruct A destruct B

9. Выберите верное утверждение:

- a) закрытые элементы базового класса в производном классе недоступны
 b) обращение к закрытым элементам класса осуществляется только через методы производных классов
 c) открытые элементы класса доступны только производным классам
 d) спецификатор доступа может обозначаться с помощью ключевых слов private, public, virtual, protected

10. Выберите верное утверждение:

- a) Указание ключа доступа необязательно, по умолчанию для классов используется ключ доступа public

- b) Производный класс наследует из базового класса поля и методы, а также деструктор, конструктор и операцию присваивания
- c) защищенные элементы при наследовании с ключом доступа `private` становятся закрытыми, а в остальных случаях доступ к ним не меняется
- d) доступ к открытым элементам остается открытым вне зависимости от ключа доступа

11. Выберите верное утверждение:

- a) с помощью перегрузки можно создавать новые операции и операторы
- b) обращение к виртуальным функциям возможно только через указатели на экземпляр класса
- c) в перегрузке функций участвуют параметры и тип перегруженной функции
- d) при использовании перегрузки операции можно полностью изменить смысл операции

12. Изучите программный код и сделайте выбор среди приведенных вариантов оценки

```
class A {
int x;
public:
    A () {cout<<"x-?"; cin>>x;}
    void SetX (int _x) { x = _x; }
};
int main () {
A *a;
SetX (10);
}
```

- a) для доступа следует писать `a -> SetX (10);`
- b) для доступа следует писать `A.SetX (10);`
- c) для доступа следует писать `a.SetX (10);`
- d) для доступа следует писать `A -> SetX (10);`

Программа зачета

1. Поиск и замена элементов строки. Алгоритмы поиска и замены.
2. Выделение подстрок. Алгоритм выделения подстроки.
3. Алгоритмы сортировки.
4. Бинарное дерево. Добавление в дерево и поиск в дереве. Удаление элемента из дерева.
5. Рекурсивные алгоритмы.
6. Статические и динамические структуры данных.
7. Стек и очередь, организация и использование.
8. Алгоритмы обработки списков.
9. Парсинг текстового файла. Форматы структурированного хранения текстовых данных.
10. Ключевые принципы объектно-ориентированного программирования
11. Определение класса (поля, методы, объекты)
12. Спецификаторы доступа
13. Правила объявления и вызова конструкторы, деструктор
14. Виды конструкторов: по умолчанию, с параметрами, копии, со списком инициализации
15. Правила объявления и вызова конструктора производного класса

16. Указатель this
17. Перегрузка операций
18. Простое наследование. Ключи доступа
19. Объявление конструкторов и деструкторов в производном классе
20. Множественное наследование
21. Виртуальные функции (объявление, определение, вызов)
22. Абстрактный класс
23. Шаблоны
24. Контейнеры

6.3 Оценочные средства для проверки уровня сформированности компетенций ОПК-

6

Тест содержит следующие типы заданий

Тип задания	№ задания	Вес задания (балл)	Результат оценивания (баллы, полученные за выполнение задания / характеристика правильности ответа)
задания закрытого типа с выбором одного правильного	1, 2, 3, 4, 5	1 балл	1 б - полное правильное соответствие; 0 б - остальные случаи
задания закрытого типа с выбором нескольких правильных ответов	6, 7	2 балла	2 б – полное правильное соответствие (последовательность вариантов ответа может быть любой); 1 б – если допущена одна ошибка / ответ правильный, но не полный; 0 б – остальные случаи
задания закрытого типа с выбором одного правильного ответа по схеме: «верно»/ «неверно»	8, 9, 10	1 балл	1 б - полное правильное соответствие; 0 б - остальные случаи
задания закрытого типа на установление соответствия	11	2 балла	2 б – полное правильное соответствие; 1 б – если допущена одна ошибка / ответ правильный, но не полный; 0 б – остальные случаи
задания открытого типа с кратким ответом	12, 13	3 балла	3 б – полное правильное соответствие; 0 б – остальные случаи.
задания открытого типа с развернутым ответом	14, 15	5 баллов	5 б – полное правильное соответствие; если допущена одна ошибка/неточность / ответ правильный, но не полный - 3 балла; если допущено более одной ошибки / ответ неправильный / ответ отсутствует – 0 баллов

Формируемая компетенция	Индикаторы сформированности компетенции
ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	ИД-1опк-6-знать: основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий; ИД-2опк-6-уметь: применять языки программирования и работы с базами данных, современные про-

	<p>граммные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ;</p> <p>ИД-3опк-6-иметь навыки: программирования, отладки и тестирования прототипов программно-технических комплексов задач</p>
--	---

Задание 1

Внимательно прочитайте задание и укажите один правильный вариант ответа:

Какой оператор в C++ используется для динамического выделения памяти под массив целых чисел?

1. malloc
2. new
3. alloc
4. create

Ответ: 2

Задание 2

Внимательно прочитайте задание и укажите один правильный вариант ответа:

Какой режим открытия файла позволяет дописывать данные в конец существующего файла?

1. ios::in
2. ios::out
3. ios::app
4. ios::trunc

Ответ: 3

Задание 3

Внимательно прочитайте задание и укажите один правильный вариант ответа:

Какой принцип ООП позволяет скрыть детали реализации класса?

1. Наследование
2. Полиморфизм
3. Инкапсуляция
4. Абстракция

Ответ: 3

Задание 4

Верно ли следующее утверждение?

В C++ для освобождения динамически выделенной памяти под массив используется оператор delete[].

Ответ: верно

Задание 5

Верно ли следующее утверждение?

Класс ifstream предназначен для записи данных в файл.

Ответ: неверно

Задание 6

Внимательно прочитайте задание и укажите три правильных варианта ответа:

Какие из перечисленных действий выполняются при работе с динамическими массивами в C++?

1. Выделение памяти
2. Инициализация элементов
3. Компиляция программы
4. Освобождение памяти
5. Объявление пространства имён
6. Вывод на экран

Ответ: 1, 2, 4

Задание 7

Внимательно прочитайте задание и укажите три правильных варианта ответа:

Какие из перечисленных элементов являются обязательными при определении класса в C++?

1. Имя класса
2. Поля (члены данных)
3. Методы (функции-члены)
4. Виртуальные функции
5. Дружественные функции
6. Шаблоны

Ответ: 1, 2, 3

Задание 8

Прочитайте текст и установите соответствие между понятием и его описанием в C++:

Динамический массив : Массив, память под который выделяется во время выполнения программы

Конструктор : Метод, вызываемый при создании объекта

Наследование : Создание нового класса на основе существующего

Поток (stream) : Абстракция для работы с вводом-выводом

Задание 9

Соотнесите режим открытия файла с его описанием:

ios::in : Открытие для чтения

ios::out : Открытие для записи (с удалением старого содержимого)

ios::app : Открытие для записи в конец файла

ios::binary : Открытие в бинарном режиме

Задание 10

Прочитайте текст и установите последовательность:

Расположите в правильной последовательности этапы работы с динамическим массивом.

Выделение памяти

Инициализация элементов

Обработка данных

Освобождение памяти

Задание 11

Прочитайте текст и установите последовательность:

Расположите в правильной последовательности выполнение операторов при работе с файлом в C++.

Открытие файла

Проверка успешности открытия

Чтение/запись данных

Закрытие файла

Задание 12

Внимательно прочитайте задание и впишите правильный ответ:

Какой оператор в C++ используется для освобождения памяти, выделенной под один объект?

Ответ: delete

Задание 13

Внимательно прочитайте задание и впишите правильный ответ:

Как называется метод класса, который не может изменять поля объекта?

Ответ: константный метод (или метод с модификатором const)

Задание 14

Внимательно прочитайте задание и запишите развернутый обоснованный ответ:

Напишите фрагмент кода на C++, который создаёт динамический массив из N целых чисел, заполняет его случайными значениями и находит сумму всех элементов.

Ответ:

```

cpp
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;
int main()
{
    int N;
    cout << "Введите размер массива: ";
    cin >> N;
    int* arr = new int[N]; // выделение памяти
    srand(time(0));

    int sum = 0;

```

```

for (int i = 0; i < N; i++) {
    arr[i] = rand() % 100; // заполнение случайными числами
    sum += arr[i];
}
cout << "Сумма элементов: " << sum << endl;
delete[] arr; // освобождение памяти
return 0;
}

```

Задание 15

Внимательно прочитайте задание и запишите развернутый обоснованный ответ:

Опишите, как в C++ организовать класс Student с полями name (строка) и grade (целое число). Реализуйте методы для установки значений, вывода данных и сохранения объекта в текстовый файл. Приведите код класса с комментариями.

Ответ:

```

cpp
#include <iostream>
#include <fstream>
#include <string>
using namespace std;
class Student {
private:
    string name;
    int grade;
public:
    // Метод для установки данных
    void setData(string n, int g) {
        name = n;
        grade = g;
    }
    // Метод для вывода данных
    void display() {
        cout << "Студент: " << name << ", Оценка: " << grade << endl;
    }
    // Метод для сохранения в файл
    void saveToFile(string filename) {
        ofstream file(filename, ios::app); // открытие для дополнения
        if (file.is_open()) {
            file << name << " " << grade << endl;
            file.close();
            cout << "Данные сохранены в файл: " << filename << endl;
        } else {
            cout << "Ошибка открытия файла!" << endl;
        }
    }
};
// Пример использования:
int main() {
    Student s;
    s.setData("Иванов", 5);
}

```

```

s.display();
s.saveToFile("students.txt");
return 0;
}

```

7 ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ В ПРОЦЕССЕ ОБУЧЕНИЯ

Информационные технологии – обучение в электронной образовательной среде с целью расширения доступа к образовательным ресурсам, увеличения контактного взаимодействия с преподавателем, построения индивидуальных траекторий подготовки, объективного контроля и мониторинга знаний студентов.

В образовательном процессе по дисциплине используются следующие информационные технологии, являющиеся компонентами Электронной информационно-образовательной среды БГПУ:

- Официальный сайт БГПУ;
- Корпоративная сеть и корпоративная электронная почта БГПУ;
- Система электронного обучения ФГБОУ ВО «БГПУ»;
- Система тестирования на основе единого портала «Интернет-тестирования в сфере образования www.i-exam.ru»;
- Система «Антиплагиат.ВУЗ»;
- Электронные библиотечные системы;
- Мультимедийное сопровождение лабораторных работ;
- система программирования Dev-C++.

8 ОСОБЕННОСТИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ ИНВАЛИДАМИ И ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

При обучении лиц с ограниченными возможностями здоровья применяются адаптивные образовательные технологии в соответствии с условиями, изложенными в раздел «Особенности организации образовательного процесса по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья» основной образовательной программы (использование специальных учебных пособий и дидактических материалов, специальных технических средств обучения коллективного и индивидуального пользования, предоставление услуг ассистента (помощника), оказывающего обучающимся необходимую техническую помощь и т.п.) с учётом индивидуальных особенностей обучающихся.

9 СПИСОК ЛИТЕРАТУРЫ И ИНФОРМАЦИОННЫХ РЕСУРСОВ

9.1 Литература

1. Алутина, Е.Ф. Лабораторный практикум по С++: учебно-методическое пособие для студентов вузов: в 2-х ч. - / Е.Ф. Алутина. – Благовещенск: Изд-во БГПУ, 2009. – Ч. 1. - 106 с. (15 экз.)
2. Алутина, Е.Ф. Лабораторный практикум по С++: учебно-методическое пособие для студентов вузов: в 2-х ч. / Е.Ф. Алутина. – Благовещенск: Изд-во БГПУ, 2009. – Ч. 2. – 104 с. (15 экз.)
3. Аляев, Юрий Александрович. Алгоритмизация и языки программирования Pascal, С++, VisualBasic [Текст]: учеб.-справочное пособие / Ю. А. Аляев, О. А. Козлов. - М.: Финансы и статистика, 2007. - 318, [1] с. (11 экз.)

4. Давыдов, В.Г. Программирование и основы алгоритмизации : Учеб. пособие для студ.вузов, обучающихся по спец."Управление и информатика в технических системах / Давыдов В.Г. – М.: Высшая школа, 2003. – 448 с. (12 экз.)
5. Иванова, Г. С. Технология программирования : учебник для студ. вузов / Г. С. Иванова. - М. : КНОРУС, 2011. - 333 с. (6 экз.)
6. Ишкова, Э.А. С++ начала программирования: учебник / Ишкова Э.А. – 3-е изд., испр.и доп. – М.: ООО «Бином-Пресс», 2004. – 368 с. (17 экз.)
7. Павловская, Т.А. С/С++ Программирование на языке высокого уровня: Учебник для студ. вузов по спец. «Информатика и вычислительная техника» /Т.А. Павловская. – СПб.и др.: Питер, 2005. – 460 с.(13 экз.).
8. Павловская, Т.А. С/С++ Программирование на языке высокого уровня: Структурное программирование: Практикум : Учебное пособие для студ. вузов по спец. «Информатика и вычисл. техника» / Т.А. Павловская, Ю.А. Щупак. – СПб.: Питер, 2002. – 238 с. (26 экз.)
9. Павловская, Т. Программирование на языке С++. – Национальный открытый университет «Интуит». Режим доступа: <http://www.intuit.ru/studies/courses/626/482/info>
10. С/С++. Алгоритмы и приемы программирования / Тимофеева В.,ред. – М.: Бином, 2003. - 560 с. (5 экз.)
11. Сальников, Ю. Н. Программирование. Базовый курс : учеб. пособие / Ю.Н. Сальников. – М. : Маркет ДС, 2010. - 335 с. (14 экз.)
12. Кувшинов, Д. Р. Основы программирования : учебное пособие для вузов / Д. Р. Кувшинов. — Москва : Издательство Юрайт, 2022. — 104 с. — (Высшее образование). — ISBN 978-5-534-07559-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/493460> (дата обращения: 13.10.2022).

9.2 Базы данных и информационно-справочные системы

1. Всероссийский образовательный портал «Информационно-коммуникационные технологии педагогам» - <https://edu-ikt.ru/>
2. Федеральный портал «Российское образование» - <http://www.edu.ru>.
3. Сайт Федеральной службы по интеллектуальной собственности, патентам и товарным знакам (Роспатента). - Режим доступа: <http://www.fips.ru/rospatent/index.htm>.

9.3 Электронно-библиотечные ресурсы

- 1.ЭБС «Юрайт». - Режим доступа: <https://urait.ru>
2. Полпред (обзор СМИ). - Режим доступа: <https://polpred.com/news>

10 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА

Для проведения лекционных и лабораторных занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации используются аудитории, оснащённые учебной мебелью, аудиторной доской, компьютером с установленным лицензионным специализированным программным обеспечением, с выходом в электронно-библиотечную систему и электронную информационно-образовательную среду БГПУ, мультимедийными проекторами, экспозиционными экранами, учебно-наглядными пособиями (мультимедийные презентации).

Для проведения лабораторных занятий также используются компьютерные классы, укомплектованные следующим оборудованием:

- Комплект столов письменных
- Аудиторная доска

- Компьютеры с установленным лицензионным специализированным программным обеспечением
- Мультимедийный проектор
- Экспозиционный экран
- Учебно-наглядные пособия - мультимедийные презентации по дисциплине «Технология программирования»

Самостоятельная работа студентов организуется в аудиториях оснащенных компьютерной техникой с выходом в электронную информационно-образовательную среду вуза, в специализированных лабораториях по дисциплине, а также в залах доступа в локальную сеть БГПУ.

Лицензионное программное обеспечение: операционные системы семейства Windows, Linux; офисные программы Microsoft office, Libreoffice, OpenOffice; Adobe Photoshop, Matlab, DrWeb antivirus и т.д.

Разработчик: Алутина Е.Ф., кандидат физико-математических наук, доцент

11 ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ

Утверждение изменений и дополнений в РПД для реализации в 20__/20__ уч. г.

РПД обсуждена и одобрена для реализации в 20__/20__ уч. г. без изменений на заседании кафедры информатики и методики преподавания информатики (протокол №__ от «__» _____ 20__ г.). В РПД внесены следующие изменения и дополнения:

№ изменения: 1	
№ страницы с изменением:	
Исключить:	Включить:
№ изменения: 2	
№ страницы с изменением:	
Исключить:	Включить: