

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Щёкина Вера Витальевна
Должность: Ректор
Дата подписания: 15.09.2023 20:29:49
Уникальный программный ключ:
a2232a55157e5f6551a8999811908e7a5398994904101000114445798



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-математи-
ческого образования и технологии
ФГБОУ ВО БГПУ

Н.В. Слесаренко

«03» сентября 2024 г.

Рабочая программа дисциплины ПРОГРАММИРОВАНИЕ

Направление подготовки

**02.03.03 МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ И
АДМИНИСТРИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ**

Профиль

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

Уровень высшего образования
БАКАЛАВРИАТ

Принята
на заседании кафедры информатики
и методики преподавания информатики
(протокол № 8 от «25» мая 2024 г.)

Благовещенск 2024

СОДЕРЖАНИЕ

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
2 УЧЕБНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ	5
3 СОДЕРЖАНИЕ ТЕМ (РАЗДЕЛОВ)	8
4 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ (УКАЗАНИЯ) ДЛЯ СТУДЕНТОВ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ	11
5 ПРАКТИКУМ ПО ДИСЦИПЛИНЕ	15
6 ДИДАКТИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ КОНТРОЛЯ (САМОКОНТРОЛЯ) УСВОЕННОГО МАТЕРИАЛА.....	15
7 ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ	32
В ПРОЦЕССЕ ОБУЧЕНИЯ	32
8 ОСОБЕННОСТИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ ИНВАЛИДАМИ И ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТАМИ ЗДОРОВЬЯ	32
9 СПИСОК ЛИТЕРАТУРЫ И ИНФОРМАЦИОННЫХ РЕСУРСОВ	33
10 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА	34
11 ЛИСТ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ	35

1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1 Цель дисциплины: формирование практических навыков применения алгоритмизации вычислительных процессов и программирования для решения экономических, вычислительных и других задач, ознакомление студентов с различными парадигмами проектирования и разработки программного обеспечения, формирование общего представления об эффективности алгоритмов и начального представления об анализе эффективности программ.

1.2 Место дисциплины в структуре ООП: Дисциплина «Программирование» относится к дисциплинам обязательной части (части, формируемой участниками образовательных отношений) блока Б1 (Б1.О.20).

Для освоения дисциплины «Программирование» используются знания, умения и виды деятельности, формируемые в процессе изучения дисциплин «Введение в направление», «Информатика», «Теоретические основы информатики», «Математический анализ», «Дискретная математика», «Алгебра и теория чисел». Дисциплина «Программирование» в профессиональной подготовке выпускника является базовой для изучения других дисциплин, связанных с программированием, изучение которой позволит студентам выработать современный подход к качеству и содержанию компьютерных программ.

Дисциплина «Программирования» обеспечивает раскрытие общего круга вопросов разработки программного обеспечения, алгоритмов, организации научных вычислений и моделирования. В ходе изучения дисциплины разбираются основные вычислительные алгоритмы, алгоритмы обработки нечисловых данных, студенты получают общее представление об эффективности алгоритмов и начальные представления об анализе их эффективности.

1.3 Дисциплина направлена на формирование следующих компетенций: ОПК-1, ОПК-6, ПК-1.

- **ОПК-1.** Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности, **индикаторами** достижения которой является:

- ОПК-1.2 – умеет использовать их в профессиональной деятельности;
- ОПК-1.3 – имеет навыки выбора методов решения задач профессиональной деятельности на основе теоретических знаний.

- **ОПК-2.** Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности, **индикаторами** достижения которой является:

- ОПК-2.1 – знает математические основы программирования и языков программирования, организации баз данных и компьютерного моделирования; математические методы оценки качества, надежности и эффективности программных продуктов; математические методы организации информационной безопасности при разработке и эксплуатации программных продуктов и программных комплексов.

- ОПК-2.2 – имеет навыки применения данного математического аппарата при решении конкретных задач.

- **ОПК-6.** Способен использовать в педагогической деятельности научные основы знаний в сфере информационно-коммуникационных технологий, **индикаторами** достижения которой является:

- ОПК-6.1 – знает изучаемый язык программирования, сетевые технологии, применение вебтехнологий.
- ОПК-6.2 – умеет вести устную и письменную коммуникации на изучаемом языке.

- **ПК-1.** Способен демонстрировать базовые знания математических и естественных наук, программирования и информационных технологий, **индикаторами** достижения которой является:

- ПК-1.1 – **знает:** обладает базовыми знаниями, полученными в области математических и (или) естественных наук, программирования и информационных технологий;
- ПК-1.2 – **умеет:** находить, формулировать и решать стандартные задачи в собственной научно-исследовательской деятельности в области программирования и информационных технологий;

1.4 Перечень планируемых результатов обучения.

В результате изучения дисциплины студент должен

знать:

- различные системы программирования на алгоритмическом языке высокого уровня C/C++;
- процесс подготовки и решения задач на ЭВМ;
- принципы, базовые концепции технологий программирования;
- основные этапы и принципы создания программного продукта;
- принципы автономной отладки и тестирования простых программ;

уметь:

- профессионально грамотно сформулировать задачу программирования;
- разрабатывать алгоритмы решения задач;
- 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;

владеть:

- языками процедурного и объектно-ориентированного программирования;
- навыками одной из технологий программирования.

1.5 Общая трудоемкость дисциплины «Программирование» составляет 9 зачетных единиц (далее – ЗЕ) (324 часа):

Программа предусматривает изучение материала на лекциях и лабораторных 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 1	Семестр 2	Семестр 3
Общая трудоемкость	432	108	216	108
Аудиторные занятия	220	66	92	62
Лекции	62	26	26	10
Лабораторные занятия	158	40	66	52
Самостоятельная работа	176	42	88	46

Экзамен	36		36	
Вид итогового контроля		Зачёт	Экзамен	Зачёт

2 УЧЕБНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

2.1 Очная форма обучения

Учебно-тематический план

№	Наименование тем (разделов)	Всего часов	Аудиторные занятия		Самостоятельная работа
			Лекции	Лабораторные занятия	
1.	Введение в предмет. Основные этапы компьютерного решения задач, критерии качества программы, диалоговые программы, дружественность. Постановка задачи и спецификация программы; способы записи алгоритма.	6	4		2
2.	Программа на языке высокого уровня. Стандартные типы данных.	22	4	8	10
3.	Структуры данных. Базовые управляющие конструкции. Представление основных структур: итерации, ветвления, повторения. Массивы данных.	80	18	32	30
	Зачет				
	Итого за 1 семестр	108	26	40	42
	2 семестр				
4.	Динамические данные. Списки: основные виды и способы реализации.	24	4	8	12
5.	Строки.	20	4	6	10
6.	Функции: построение и использование.	20	4	6	10
7.	Программирование рекурсивных алгоритмов.	14	2	6	6
8.	Способы конструирования программ; модульные программы.	30	2	14	14

9.	Типы данных, определяемые пользователем; структуры.	54	6	22	26
10.	Анализ правильности программ, основы доказательства правильности.	12	2	4	6
11.	Архитектура и возможности семейства языков высокого уровня.	6	2	0	4
	Экзамен	36			
	Итого за 2 семестр	216	26	66	88
	3 семестр				
12.	Типы данных, определяемые пользователем; файлы. Текстовые, бинарные файлы.	20	2	10	8
13.	Принципы объектно-ориентированного подхода к разработке программ. Определение класса: поля, методы, объекты.	32	4	16	12
14.	Инкапсуляция и спецификация правил доступности элементов классов. Наследование.	23	1	12	10
15.	Виртуальные функции.	9	1	4	4
16.	Перегрузка функций и операций.	8		4	4
17.	Шаблоны.	10	2	6	4
18.	Контейнеры.	6		2	4
	Итого за 3 семестр	108	10	52	46
	ИТОГО	432	62		

Интерактивное обучение по дисциплине

№	Наименование тем (разделов)	Вид занятия	Форма интерактивного занятия	Кол-во часов
1.	Программа на языке высокого уровня. Стандартные типы данных.	Лек	Презентации с использованием вспомогательных средств с обсуждением	2

2.	Структуры данных. Базовые управляющие конструкции. Представление основных структур: ветвления.	Лек	Лекция с ошибками Презентации с использованием вспомогательных средств с обсуждением	2
3.	Структуры данных. Базовые управляющие конструкции. Представление основных структур: итерации.	Лек	Лекция с ошибками	2
4.	Структуры данных. Базовые управляющие конструкции. Представление основных структур: повторения.	Лек	Лекция - дискуссия	2
5.	Программа на языке высокого уровня. Стандартные типы данных.	Лб.	Разминка	2
6.	Программа на языке высокого уровня. Стандартные типы данных.	Лб.	Работа в малых группах	2
7.	Структуры данных. Базовые управляющие конструкции. Представление основных структур: итерации, ветвления, повторения.	Лб.	Разминка	2
8.	Структуры данных. Базовые управляющие конструкции. Представление основных структур: итерации, ветвления, повторения.	Лб.	Работа в малых группах	2
9.	Структуры данных. Базовые управляющие конструкции. Представление основных структур: итерации, ветвления, повторения.	Лб.	Разминка	2
10.	Структуры данных. Базовые управляющие конструкции. Представление основных структур: итерации, ветвления, повторения.	Лб.	Работа в малых группах	2
11.	Структуры данных. Базовые управляющие конструкции. Представление основных структур: итерации, ветвления, повторения.	Лб.	Разминка	2
12.	Структуры данных. Базовые управляющие конструкции. Представление основных структур: итерации, ветвления, повторения.	Лб.	Работа в малых группах	2
13.	Динамические структуры данных. Списки: основные виды и способы реализации.	Лек	Презентации с использованием вспомогательных средств с обсуждением	2
14.	Динамические структуры данных. Списки: основные виды и способы реализации.	Лб.	Разминка	2
15.	Динамические структуры данных. Списки: основные виды и способы реализации.	Лб.	Работа в малых группах	2

16.	Функции: построение и использование.	Лек	Лекция с ошибками Презентации с использованием вспомогательных средств с обсуждением	2
17.	Функции: построение и использование.	Лб	Разминка	2
18.	Функции: построение и использование.	Лб	Работа в малых группах	2
19.	Программирование рекурсивных алгоритмов.	Лек	Обратная связь	2
20.	Программирование рекурсивных алгоритмов.	Лб	Разминка	2
21.	Программирование рекурсивных алгоритмов.	Лб	Работа в малых группах	2
22.	Типы данных, определяемые пользователем; записи; файлы.	Лек	Обратная связь Презентации с использованием вспомогательных средств с обсуждением	2
23.	Типы данных, определяемые пользователем; записи; файлы.	Лб	Разминка	2
24.	Способы конструирования программ; модульные программы.	Лб	Работа в малых группах	2
ИТОГО				48

3 СОДЕРЖАНИЕ ТЕМ (РАЗДЕЛОВ)

1. ВВЕДЕНИЕ В ПРЕДМЕТ. ОБЩИЕ ПОНЯТИЯ О ПРОГРАММЕ. СТРУКТУРА ПРОГРАММЫ. СПОСОБЫ КОНСТРУИРОВАНИЯ ПРОГРАММ

Место дисциплины «Программирование» в системе подготовки студентов по направлению подготовки 02.03.03 «Математическое обеспечение и администрирование информационных систем». Программа, цели и задачи дисциплины. Введение в предмет. История развития программирования как науки и искусства. История развития программного обеспечения. Проектирование программ. Парадигмы программирования. Методологии программирования. Развитие технологий программирования. Разработка программ методом пошаговой детализации. Программирование как процесс построения моделей объектов реального мира и организации взаимодействия между ними.

Диалоговые программы, дружественность. Основные этапы решения задач на ЭВМ. Постановка задачи и спецификация программы. Способы записи алгоритма, программа на языке высокого уровня. Общие понятия о программе, исполнении, среде программирования. Виды трансляторов: интерпретаторы и компиляторы. Общая схема работы транслятора. Процесс компиляции программы. Виды ошибок: синтаксические, семантические.

2. ПРОГРАММА НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ. СТАНДАРТНЫЕ ТИПЫ ДАННЫХ

История появления языка Си и С++. Структура программы на языке высокого уровня С++. Директивы препроцессора. Понятия о синтаксисе и семантике формального языка. Нормальные формы Бэкуса-Наура и синтаксические диаграммы Вирта. Основные понятия языка: алфавит языка программирования, лексемы, идентификаторы, служебные (ключевые) слова, литералы, выражения, управляющие символы (escape-последовательности),

операции, разделители языка. Операции инкремент, декремент. Префиксная и постфиксная формы записи. Концепция типов данных в алгоритмических языках. Стандартные типы данных и операции с ними. Простые типы. Структурные типы. Иерархия типов, используемых в языках высокого уровня. Базовые типы: char, int, float, double, long double. Модификаторы типов. Диапазоны значений данных. Операции применимые с базовыми типами данных, особенности операции деления. Библиотека математических функций.

3. СТРУКТУРЫ ДАННЫХ. БАЗОВЫЕ УПРАВЛЯЮЩИЕ КОНСТРУКЦИИ. ПРЕДСТАВЛЕНИЕ ОСНОВНЫХ СТРУКТУР: ИТЕРАЦИИ, ВЕТВЛЕНИЯ, ПОВТОРЕНИЯ

Значения и типы. Константы и переменные. Описания и операторы. Присваивание – оператор или операция. Дополнительные операции присваивания. Операция явного приведения типов. Неявное приведение типов. Форматированный ввод – вывод. Потоковый ввод – вывод. Заголовочные файлы библиотек. Спецификаторы формата. Манипуляторы. Принципы структурного программирования. Основные алгоритмические структуры и их суперпозиции. Структуризация операторов (циклы, ветвления). Операции отношения, логические операции. Организация ветвлений: операция условие, условный оператор, оператор выбора. Полное и неполное ветвление. Условные выражения. Вложенные условные операторы. Оператор выхода break. Пустой оператор. Составной оператор. Организация повторений. Цикл с пред и постусловием, счетный цикл. Числовые массивы. Алгоритмы поиска элементов массива. Сортировки: выбором, обменом, вставкой. Анализ сложности алгоритмов на примере сортировок.

4. ДИНАМИЧЕСКИЕ СТРУКТУРЫ ДАННЫХ. СПИСКИ: ОСНОВНЫЕ ВИДЫ И СПОСОБЫ РЕАЛИЗАЦИИ

Указатели. Динамические переменные. Процедуры создания и уничтожения данных. Ссылки. Указатели и массивы. Динамическое распределение памяти. Выделение динамической памяти под массивы.

5. СТРОКИ

Массивы символов. Библиотека обработки строк String, стандартные функции обработки строковых данных.

6. ФУНКЦИИ: ПОСТРОЕНИЕ И ИСПОЛЬЗОВАНИЕ

Представление данных и знаний с помощью процедур (функций). Понятие о функциональном программировании. Формальные и фактические параметры. Локальные и глобальные переменные. Область действия и время жизни программных объектов. Передача параметров по значению, через глобальные переменные, по ссылке.

7 ПРОГРАММИРОВАНИЕ РЕКУРСИВНЫХ АЛГОРИТМОВ

Рекурсивные функции: описание работы с помощью пред- и постусловий, неформальное доказательство правильности работы рекурсивной функции по индукции.

8. ТИПЫ ДАННЫХ, ОПРЕДЕЛЯЕМЫЕ ПОЛЬЗОВАТЕЛЕМ: СТРУКТУРЫ, ФАЙЛЫ

Файлы. Понятие логического и физического файлов. Файловые типы. Общие функции для работы с файлами. Типизированные файлы. Текстовые и бинарные файлы. Процедуры ввода-вывода данных в файлы. Прямой и последовательный доступ к компонентам файлов.

9. СПОСОБЫ КОНСТРУИРОВАНИЯ ПРОГРАММ; МОДУЛЬНЫЕ ПРОГРАММЫ

Классификация по ядрам методологии: императивное программирование, объектно-ориентированное, функциональное, логическое. 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).

10. АНАЛИЗ ПРАВИЛЬНОСТИ ПРОГРАММ, ОСНОВЫ ДОКАЗАТЕЛЬСТВА ПРАВИЛЬНОСТИ

Понятие о состоянии. Программа как преобразователь состояний. Множества состояний и их описание с помощью логических функций (предикатов). Понятие о пред- и постусловиях, заданных предикатами. Правильность программы как функция предусловия, постусловия. Исчисление программ. Слабейшее предусловие. Нахождение слабейшего предусловия как основная цель доказательства правильности программ. Основные аксиомы и правила вывода. Инвариант и теорема о цикле. Построение программы одновременно с доказательством.

11. АРХИТЕКТУРА И ВОЗМОЖНОСТИ СЕМЕЙСТВА ЯЗЫКОВ ВЫСОКОГО УРОВНЯ

История развития и эволюция языков программирования. Исторически сложившиеся подходы в программировании ЭВМ. Номенклатура языков программирования. Программирование, управляемое данными. Программирование, управляемое событиями. Поддержка событийного программирования операционной средой. Диспетчер сообщений. Очередь сообщений. Генераторы и обработчики сообщений.

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. Дидактические материалы для контроля (самоконтроля) усвоения учебного материала содержат вопросы допуска к лабораторным работам, примерные вопросы зачета и экзамена, а так же индивидуальные и тестовые задания. 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;
- краткие теоретические и учебно-методические материалы по каждой теме, позволяющие студенту ознакомиться с вопросами, обсуждаемыми на лабораторных работах;
- 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 лекциям

Курс лекций строится на основе четких понятий и формулировок, так, как только при таком подходе студенты приобретают культуру абстрактного мышления, необходимую для высококвалифицированного бакалавра в любой отрасли знаний, а также на разборе задач и алгоритмов их решения. Изложение материала должно быть по возможности простым и базироваться на уровне разумной строгости. Изложение теоретического материала дисциплины должно предшествовать лабораторным занятиям.

4.3 Методические рекомендации по подготовке к лабораторным занятиям

Целью лабораторных занятий является выработка у студентов навыков практической работы на компьютере, выработка у них начальных навыков алгоритмизации, умения составлять простейшие компьютерные программы на одном из алгоритмических языков. Целью лабораторных работ является закрепление теоретического материала лекций и выработка умения решать задачи.

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

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

Для подготовки ответов на вопросы допуска необходиима подготовка по материалам лекций, а также чтение дополнительной литературы. 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.

Для решения большей части заданий требуются знания из различных разделов математики. Задачи для индивидуальных заданий подобраны на основе сборников задач по программированию (авторы В.Н. Пильщикова, С.А. Абрамов) и других источников.

Первый вариант домашнего задания должен быть выполнен письменно в тетради, второй вариант предлагается для обсуждения в группе, подобные задания включены в тесты на проверку самостоятельной работы студента.

Варианты индивидуальных заданий назначаются преподавателем. Индивидуальные задания выполняются студентом дома или на занятиях самоподготовки и сдаются на следующем занятии с подробным объяснением каждой строки кода программы. 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;
- ответы на вопросы допуска соответствующей лабораторной работы;
- тексты программ или решения заданий 1 варианта домашнего задания;
- результаты отладки (ошибки, их причины и способы устранения);
- результаты работы программы для различных исходных данных.

Условием сдачи зачета за 1 семестр является выполнение домашних и 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 качестве образца решения задач следует брать те решения, которые приводились преподавателем на лекциях или выполнялись на лабораторных занятиях. В случае появления каких-либо вопросов следует обращаться к преподавателю в часы его консультаций. Критерием качества усвоения знаний могут служить аттестационные оценки по дисциплине и текущие оценки, выставляемые 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. Одной из форм интерактивных форм проведения занятия является дискуссия.

Важно предварительно определить правила ведения дискуссии, а также критерии оценки выступлений ее участников. Лучше всего это сделать предметом обсуждения в группе, а не предлагать преподавателем в готовом виде. Например, обсудить и принять следующие или похожие правила ведения дискуссии:

- не допускать выпадов против личности;
- не допускать излишнюю эмоциональность;
- высказываться четко, кратко и по теме обсуждения;

- выбрать ведущего дискуссии, ответственного за время и правила ее проведения;
- дать высказаться всем желающим, уважительно относиться к любой точке зрения;
- внимательно слушать друг друга, не говорить одновременно;
- постараться проанализировать разные точки зрения;
- не повторяться, продвигать дискуссию дальше через движение новых идей, сообщение новой информации.

Обсудить и принять в группе следующие критерии оценки участия в дискуссии:

- точность аргументов (причинно-следственные связи);
- четкость и понятность аргументации;
- точность контраргументов (причинно-следственные связи);
- четкость понятность контраргументов;
- логичность;
- удачная подача;
- умение выделить главное;
- отделение фактов от субъективного мнения;
- использование ярких поддерживающих фактов;
- видение сути проблемы;
- ориентация меняющейся ситуации.

4.4 Методические указания к самостоятельной работе студентов

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

- регулярную (еженедельную) проработку теоретического материала по конспектам лекций и учебникам;
- регулярную (еженедельную) подготовку к лабораторным занятиям, в том числе изучение описания лабораторных работ;
- регулярное (еженедельное) решение индивидуальных и домашних задач и упражнений, задаваемых преподавателем;
- подготовка к контрольной работе и ее успешное выполнение.

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

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

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

Наименование раздела (темы)	Формы/виды самостоятельной работы	Кол-во часов
Введение в предмет. Основные этапы компьютерного решения задач, критерии качества программы, диалоговые программы, дружественность. Постановка задачи и спецификация программы; способы записи алгоритма.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Решение домашних и индивидуальных задач. Подготовка отчетов по индивидуальным заданиям.	2

Программа на языке высокого уровня. Стандартные типы данных.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Решение домашних и индивидуальных задач. Подготовка отчетов по индивидуальным заданиям.	10
Структуры данных. Базовые управляющие конструкции. Представление основных структур: итерации, ветвления, повторения.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Решение домашних и индивидуальных задач. Подготовка отчетов по индивидуальным заданиям.	30
Динамические структуры данных. Списки: основные виды и способы реализации.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Решение домашних и индивидуальных задач. Подготовка отчетов по индивидуальным заданиям.	12
Строки	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Решение домашних и индивидуальных задач. Подготовка отчетов по индивидуальным заданиям.	10
Функции: построение и использование.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Решение домашних и индивидуальных задач. Подготовка отчетов по индивидуальным заданиям.	10
Программирование рекурсивных алгоритмов.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Решение домашних и индивидуальных задач. Подготовка отчетов по индивидуальным заданиям.	6
Способы конструирования программ; модульные программы.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Решение домашних и индивидуальных задач. Подготовка отчетов по индивидуальным заданиям.	14
Типы данных, определяемые пользователем: структуры, файлы.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Решение домашних и индивидуальных задач. Подготовка отчетов по индивидуальным заданиям.	26
Анализ правильности программ, основы доказательства правильности.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Решение домашних и индивидуальных задач. Подготовка отчетов по индивидуальным заданиям.	6
Архитектура и возможности семейства языков высокого уровня.	Проработка теоретического материала по конспектам лекций и материалам СЭО БГПУ. Решение домашних и индивидуальных задач. Подготовка отчетов по индивидуальным заданиям.	4
Итого		130

5 ПРАКТИКУМ ПО ДИСЦИПЛИНЕ

5. ПРАКТИКУМ ПО ДИСЦИПЛИНЕ

5.1 План проведения лабораторных занятий по дисциплине

Программа на языке высокого уровня. Стандартные типы данных.

Лабораторная работа № 1. Вводное занятие. Компиляция и отладка программы.

Лабораторная работа № 2. Линейные программы. Ввод/вывод данных. Инициализация переменных.

Структуры данных. Базовые управляющие конструкции. Представление основных структур: итерации, ветвлений, повторения. Массивы данных.

Лабораторная работа № 3. Перечислимые константы. Условные операторы.

Лабораторная работа № 4. Организация повторений. Цикл с пред- и постусловием.

Лабораторная работа № 5. Организация повторений. Цикл с параметром.

Лабораторная работа № 6. Обработка массивов.

Динамические структуры данных. Списки: основные виды и способы реализации. Строки.

Лабораторная работа № 7. Динамические массивы. Работа с указателями.

Лабораторная работа № 8. Обработка строк (символьных массивов).

Функции: построение и использование.

Лабораторная работа № 9. Использование функций.

Программирование рекурсивных алгоритмов.

Лабораторная работа № 10. Рекурсивные алгоритмы. Перегрузка функций.

Типы данных, определяемые пользователем; структуры; файлы.

Лабораторная работа № 11. Структуры.

Лабораторная работа № 12. Обработка файлов.

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-1, ОПК-6, ПК-1.	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
		Высокий (отлично)	Ставится, если продемонстрированы знание вопроса и самостоятельность мышления, ответ соответствует требованиям правильности, полноты и аргументированности.
		Низкий (неудовлетворительно)	Ответ студенту не зачитывается если: <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)	Задание выполнено более, чем на половину. Студент обнаруживает знание и понимание основных положений задания, но: <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.
ОПК-1 ОПК-2 ОПК-6 ПК-1	Разноуровневые задачи и задания (индивидуальные задания)	Базовый (хорошо)	Задание в основном выполнено. Ответы правильные, но: <ul style="list-style-type: none"> • В ответе допущены малозначительные ошибки и недостаточно полно раскрыто содержание вопроса; • Не приведены иллюстрирующие примеры, недостаточно чётко выражено обобщающие мнение студента; • Допущено 1-2 недочета в последовательности и языковом оформлении излагаемого.
		Высокий (отлично)	Задание выполнено в максимальном объеме. Ответы полные и правильные. <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;

			<ul style="list-style-type: none"> Излагает материал последовательно и правильно с точки зрения норм литературного языка.
ОПК-1, ОПК-6, ПК-1.	Лабораторная работа	Низкий (неудовлетвори- тельно)	<p>Лабораторная работа студенту не за- считывается если студент:</p> <ol style="list-style-type: none"> просто набрал примеры без измене- ний и комментарий из лабораторной работы; не ответил на теоретические вопросы к лабораторной работе или не выпол- нил самостоятельное задание; или если правильно выполнил менее половины работы.
		Пороговый (удовлетворительно)	<p>Если студент правильно выполнил не менее половины работы при этом:</p> <ol style="list-style-type: none"> попытался внести изменения в при- меры из лабораторной работы и добав- ил не верные комментарии; ответил на теоретические вопросы к лабораторной работе, но не выполнил самостоятельное задание.
		Базовый (хорошо)	<p>Если студент правильно выполнил работу полностью при этом:</p> <ol style="list-style-type: none"> допустил незначительные ошибки при ответе на теоретические вопросы к лабораторной работе; при выполнении самостоятельных заданий допустил незначительные ошибки в комментариях или оформле- нии вывода данных.
		Высокий (отлично)	<p>Если студент:</p> <ol style="list-style-type: none"> выполнил работу без ошибок и недочетов; допустил не более одного недочета.

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, экзамен.

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 75 и более процентов;

- он имеет посещаемость лабораторных занятий не менее 75 процентов.

Оценка «не зачтено» выставляется студенту, если:

- задания, размещенные в Электронной информационно-образовательной среде

БГПУ выполнены менее чем на 75 процентов;

- он имеет посещаемость лабораторных занятий менее 75 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.).

Критерии оценивания устного ответа на экзамене

В билетах сформулированы задачи на 4 и 3, условия которых аналогичны задачам, предлагаемым для индивидуального решения. При ответе на экзамене студент может поменять задачу на 5 (нестандартную задачу).

1. Экзаменационная оценка «отлично» выставляется, если студент:
 - безуказненно ответил на оба вопроса экзаменационного билета.
 - написал программу решения задачи «на 5», проверил ее выполнение для различных исходных данных;
 - в ответе допустил 1-2 недочета или одну негрубую ошибку.
2. Экзаменационная оценка «хорошо» выставляется, если студент ответил на все вопросы экзаменационного билета, но:
 - в ответе на теоретические вопросы допустил 3-5 недочета;
 - в ответе допустил одну негрубую ошибку и 1-3 недочета;
 - в ответе на теоретические вопросы допустил 2-3 негрубых ошибки;
 - при решении задачи на 5, условие задачи несколько упрощено;
 - задача на 4 работает, но возможно не учтено одно из условий, накладываемых на исходные данные.
3. Экзаменационная оценка «удовлетворительно» выставляется, если студент ответил на все вопросы экзаменационного билета, но:
 - в ответе допустил более 5 недочетов;
 - в ответе допустил более 3 негрубых ошибок;
 - в ответе допустил 1-3 грубых ошибки.
 - студент приступил к решению задачи, но не довел решение до конца, при этом объявил все необходимые данные, осуществил ввод исходных данных, вывод решения, при реализации алгоритма использовал необходимые управляющие структуры, допустил грубые ошибки в реализации алгоритма.

Экзаменационная оценка «удовлетворительно» может быть выставлена и в случае, если студент без грубых ошибок ответил на 1 вопрос экзаменационного билета и решил задачу, с недочетами.

4. Экзаменационная оценка «неудовлетворительно» выставляется, если:
 - студент ответил на все теоретические вопросы экзаменационного билета, но в ответе допустил более 3 грубых ошибок;
 - студент ответил на 2теоретических вопроса экзаменационного билета, но в ответе допустил грубые ошибки, при этом к решению задачи не приступал, т.е. не объявил необходимые для решения задачи данные и с грубыми ошибками написал управляющие структуры, необходимые для обработки исходных данных, либо использовал не ту управляющую структуру.
 - студент ответил на менее чем 2 вопроса экзаменационного билета.

Экзаменационная оценка может быть повышена, если в своем ответе студент показал знание монографической и учебной литературы, выходящей за рамки дисциплины, проявил оригинальность и высокую культуру абстрактного мышления.

К грубым ошибкам относятся:

- незнание основных понятий и управляющих структур, алгоритмов обработки данных;
- полное отсутствие детальной проработки излагаемого при наличии основных этапов изложения;

- неумение применять методы и алгоритмы дисциплины к решению задач;
- ошибки в алгоритмах, приводящие к существенному искажению окончательных результатов;
- другие ошибки, равнозначные указанным.

К негрубым ошибкам относятся:

- частичное отсутствие детальной проработки излагаемого при наличии основных этапов изложения;
- ошибки в алгоритмах, не приводящие к сильному искажению окончательных результатов;
- другие ошибки, равнозначные указанным.

К недочетам относятся:

- нерациональное решение экзаменационных задач;
- запись ответов в виде, допускающем упрощение;
- неточности в пояснениях;
- неаккуратное выполнение работы;
- другие ошибки, равнозначные указанным.

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

Примеры домашних заданий

1. Найдите ошибки в объявлениях массивов, объясните их:

- a) constfloatn=10; b) intn=5;
 inta[n]; inta[n1];
 c) intn=5;
 inta[5]={ 6, 7, 9, 11, 13, 15,
 17};

2. Что будет выведено на экран после выполнения программ?

- a) int main() b) int main()
 { int k=10; { int k=10;
 int a[k]={1, 2, 3, 4, 5}; int a[k]={1, 2, 3, 4, 5};
 cout<<a[5]<<endl; cout<<a[0]<<endl;
 } }

3. Что напечатает следующая программа?

- a) int n=10, a[10]={2, 4, 6, 8, 10};
 for (inti=5 ; i<n; i++)
 cout<<a[i]<<endl;
 b) int n=10, a[]={1, 3, 5, 7, 9, 11, 13, 15, 17,19,21};
 for (inti=0 ; i<n; i+=2)
 cout<<a[i]<<endl;

4. Вычислите значение логических выражений:

- a)pow(x,2)+pow(y,2)<=4, при x=0.3, y= -1.6
 b) k % 7 == k / 5-1, при k=15
 c) (x*y!=0) && (y>x), при x=2, y=1
 d) (x*y!=0) || (y>x), при x=2, y=1

5. Запишите выражение на C++, принимающее значение истина или ложь:

- a) целое k делится на 7;
 b) точка (x, y) лежит вне круга радиуса r с центром в точке (1,0);
 c) натуральное n является полным квадратом;
 d) x<-1 или x>=1;
 e) x = max(x, y, z);

- f)** x не принадлежит отрезку $[0,1]$;

h) каждое из чисел x, y, z отрицательно;

j) числа x, y, z равны между собой;

k) из чисел x, y, z только два равны между собой;

l) 5 входит в десятичную запись трехзначного целого числа.

6. Нарисуйте на плоскости (x,y) область, в которой и только в которой истинно указанное выражение:

 - $(y \geq x) \&& (y+x \geq 0) \&& (y \leq 1)$
 - $(\text{pow}(x,2)+\text{pow}(y,2) > 1) \parallel (y < 0) \&& (\text{fabs}(x) \leq 1)$

7. Допустимы ли следующие составные операторы? Исправьте ошибки:

 - { }
 - { $x=0$; }
 - { ; };

8. Определите значение логического выражения

$$(-3 \geq 5) \parallel !(7 < 9) \&& (0 < 3)$$

9. Запишите условный оператор, в котором значение переменной x вычисляется по формуле $a+b$, если a – нечетное и $a*b$, если a – четное.

10. Найдите синтаксические ошибки в записи команд:

 - if $x > 0$ $y = x$; Else $y = -x$;
 - if ($t > 30$) $y = 1$ else $y = 0$;
 - if ($z < 0$)
{ printf ("Меньше нуля");
elseprintf ("\nБольше или равно нулю"); }
 - if ($a < b < c$); $x = x + 1$;
 - switch (n)
{ case 1: $y = 3$;
case 2: case 4 $y = 4$;
case 3 $y = 5$;
default: printf("Неверное значение");
}
 - switch k
case 1..41: $y = 2$;
case 1..73: $y = 3$;
}

11. Запишите в виде команды выбора

```
if ( $x = -1 \parallel x = 8$ )     $k = k + 1$ ;  

if ( $x = -9 \parallel x = 12$ )     $k = k + 2$ ;  

if ( $x = -3$ )     $k = k + 3$ ;
```

12. Есть ли в следующих примерах пустые операторы?

```
if ( $x > 0$ )  $x = 1$ ; else ;  $y++$ ;  

switch (n)  

{ case 1: ;  

case 2: case 4  $y = 1$ ; }
```

13. Сколько раз выполнится цикл в следующих программах?

 - $\text{int } k = 5;$ $c) \text{ int } k = 15;$
 $\text{while } (k \leq 15) k++;$ $\text{while } (k \geq 15) k--;$
 - $\text{int } k = 15;$ $d) \text{ int } k = 15;$
 $\text{while } (k \geq 15) k++;$ $\text{while } (k > 10) k--;$

14. При каких исходных значениях t цикл будет выполнять бесконечно?

 - $\text{int } t;$ $b) \text{ int } t;$
 $\text{while } (t < 5) t--;$ $\text{while } (t > 1) t++;$

15. Перепишите код приведенной программы с использованием цикла с пост условием.

```
int main()
{ int i=1, sum=0;
while ( i<10)
{ sum+=i;
i+=2;
}
printf("Сумма=%d\n", sum);
}
```

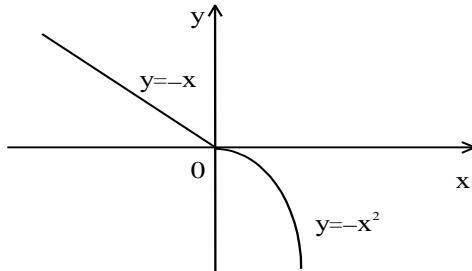
**Варианты собеседования по теме:
АРХИТЕКТУРА И ВОЗМОЖНОСТИ СЕМЕЙСТВА
ЯЗЫКОВ ВЫСОКОГО УРОВНЯ**

1. Эволюция языков программирования.
2. Развитие технологий программирования.
3. Программирование, управляемое данными. Программирование, управляемое событиями. Поддержка событийного программирования операционной средой.
4. Диспетчер сообщений. Очередь сообщений. Генераторы и обработчики сообщений.

**Варианты разноуровневых задач и заданий
Индивидуальное задание 2. Условный оператор**

Вариант 1

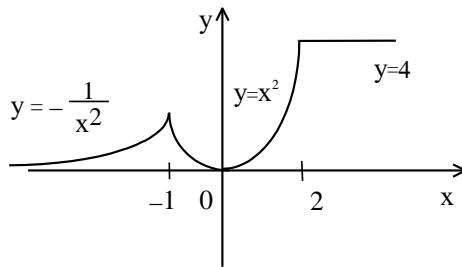
1. Дано действительное число a . Для функции $f(x)$, график которой представлен на рисунке, вычислить $f(a)$.



2. Определить, является ли введенное значение года високосным, если введено корректное значение.

Вариант 2

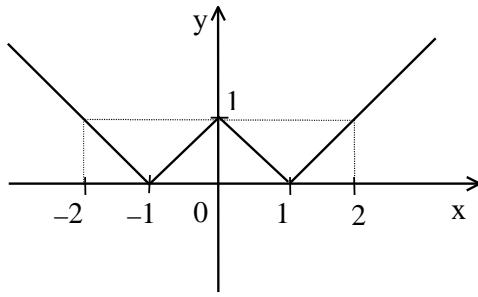
1. Дано действительное число a . Для функции $f(x)$, график которой представлен на рисунке, вычислить $f(a)$.



2. Дано натуральное число n ($n \leq 100$), определяющее возраст человека (в годах). Дать для этого числа наименования год, года, лет: например, 1 год, 23 года, 45 лет и так далее.

Вариант 3

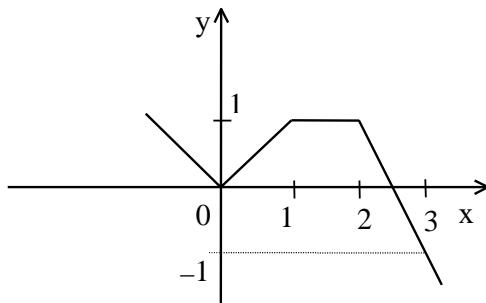
1. Дано действительное число a . Для функции $f(x)$, график которой представлен на рисунке, вычислить $f(a)$.



2. Доказать, что любую целочисленную сумму, большую 7 руб. можно выплатить без сдачи трешками и пятерками. Для данного $n > 7$ найти такие целые неотрицательные a, b , что $3a + 5b = n$.

Вариант 4

1. Дано действительное число a . Для функции $f(x)$, график которой представлен на рисунке, вычислить $f(a)$.

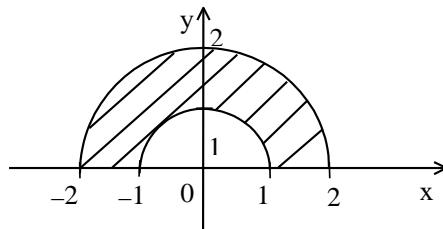


2. Поле шахматной доски определяется парой натуральных чисел, первое число – номер вертикали, второе – номер горизонтали. Даны натуральные числа k, l, m, n , каждое из которых не превосходит восьми. Требуется: выяснить, являются ли поля (k, l) и (m, n) полями одного цвета.

Вариант 5

1. Пусть D – заштрихованная часть плоскости. Даны действительные числа x, y . Определить u , если u определяется по x и y следующим образом:

$$u = \begin{cases} 0, & \text{если } (x, y) \in D \\ x, & \text{в противном случае} \end{cases}$$



2. Поле шахматной доски определяется парой натуральных чисел, первое число – номер вертикали, второе – номер горизонтали. Даны натуральные числа k, l, m, n , каждое из которых не превосходит восьми. На поле с координатами (k, l) расположен ферзь. Угрожает ли он полю с координатами (m, n) ?

Индивидуальное задание 3 Цикл с предусловием (while)

Вариант 10

1. Дано действительное число $x \neq 0$. Вычислить

$$\frac{x}{x^2 + \frac{2}{x^2 + \frac{4}{x^2 + \frac{8}{x^2 + \frac{\vdots}{x^2 + \frac{256}{x^2}}}}}}$$

2. Вычислить сумму ряда с точностью Е, общий член которого $a_n = \frac{n!}{3n^n}$.

Вариант 11

1. Вычислить $\frac{1}{1 + \frac{1}{3 + \frac{1}{5 + \frac{1}{\vdots + \frac{1}{101 + \frac{1}{103}}}}}}$

2. Вычислить сумму ряда с точностью Е, общий член которого $a_n = \frac{3n + n!}{(3n)!}$.

Вариант 12

1. Дано натуральное число n . Вычислить произведение первых n сомножителей $\frac{1}{2} * \frac{3}{4} * \frac{5}{6} * \dots$

2. Вычислить сумму ряда с точностью Е, общий член которого $a_n = e^{-n}$.

Вариант 13

1. Дано натуральное число n . Вычислить $1 \cdot 2 + 2 \cdot 3 \cdot 4 + \dots + n \cdot (n+1) \cdot \dots \cdot 2n$

2. Вычислить сумму ряда с точностью Е, общий член которого $a_n = \frac{1}{n^2}$.

Вариант 14

1. Дано натуральное число n , действительное число x . Вычислить

$$\sin x + \sin \sin x + \dots + \sin \sin \dots \sin x$$

2. Вычислить сумму ряда с точностью Е, общий член которого $a_n = \frac{3^n * n!}{(3n)!}$.

Вариант 15

1. Дано натуральное число n , действительное число x . Вычислить $\sin x + \sin x^2 + \dots + \sin x^n$

2. Вычислить сумму ряда с точностью Е, общий член которого $a_n = \frac{2^n}{\sqrt{n!}}$.

Индивидуальное задание 4. Счетный цикл

Вариант 5

- Найти количество делителей натурального числа, больших K (K вводить с клавиатуры).
- Найти все натуральные числа a, b, c из интервала от 1 до 20, для которых выполняется равенство $a^2 \cdot b = c^2$.
- Квадрат трехзначного числа оканчивается тремя цифрами, которые как раз и составляют это число. Написать программу поиска таких чисел.
- Даны натуральные числа m и n . Получить все кратные им числа, меньшие $m \cdot n$.

Вариант 6

1. Найти сумму целых чисел из промежутка от 1 до 200, у которых ровно 5 делителей.

2. Найти все такие тройки натуральных чисел x, y, z из интервала от 1 до 20, для которых выполняется равенство: $x^2+y^2=z^2$.
3. Написать программу поиска четырехзначного числа, которое при делении на 133 дает в остатке 125, а при делении на 134 дает в остатке 111.
4. Найти количество четных цифр заданного целого положительного числа.

Вариант 7

1. Найти все целые числа из промежутка от 100 до 300, у которых сумма делителей равна K (K вводится с клавиатуры).
2. Найти все такие тройки натуральных чисел x, y, z из интервала от 1 до 20, для которых выполняется равенство: $x^2+y^2-z^2=0$.
3. Найти сумму положительных нечетных чисел, меньших 100.
4. Приписать по 1 в начало и конец записи числа n . Например, было – 3456, стало – 134561.

Вариант 8

1. Найти все натуральные числа из промежутка от a до b , у которых количество делителей превышает заданное число K .
2. Найти все натуральные числа a, b, c из интервала от 1 до 20, для которых выполняется равенство $a+b=c^2$.
3. Найти сумму целых положительных чисел из промежутка от A до B , кратных 4 (значения A, B вводятся с клавиатуры).
4. Поменять порядок цифр числа на обратный. Например, было – 12345, стало – 54321.

Вариант 9

1. Найти сумму четных делителей натурального числа.
2. Найти все равновеликие прямоугольные треугольники, катеты которых выражены целыми числами a и b , а площадь равна s (a и b принадлежат интервалу от 1 до 20, а s вводится с клавиатуры).
3. Найти сумму целых положительных чисел, больших 20, меньших 100, кратных 3 и заканчивающихся на 2, 4 или 8.
4. Поменять местами первую и последнюю цифру числа. Например, было – 12345, стало – 52341.

Примеры заданий для контрольных работ

1. Сгенерировать первые K членов ряда Фибоначчи, рекуррентная формула которого $n_i=n_{i-1}+n_{i-2}$ и два первых члена равны $n_0=0, n_1=1$.
2. Выдать в три столбика нечетные числа натурального ряда, не превышающие числа N и делящиеся нацело на 3, 5 и 7.
3. Подсчитать количество нулевых элементов в последовательности из N чисел и выдать последовательность в обратном порядке.
4. В матрице размера $M \times N$ обнулить элементы, не попадающие в интервал от k_1 до k_2 , и выдать транспонированную матрицу.
5. Во вводимой последовательности из 10 чисел, начиная со второго числа, определить количество чисел больших предыдущего числа.
6. Определить, является ли автобусный билет счастливым, т.е. равны ли суммы старших и младших трех цифр шестизначного числа.
7. Для заданных чисел $a_1, b_1, c_1, a_2, b_2, c_2$ выдать координаты точки пересечения прямых, описываемых уравнениями $a_1x+b_1y=c_1$ и $a_2x+b_2y=c_2$, или сообщить, что эти прямые совпадают или параллельны.
8. По названию страны (для 10–12 стран) выдавать столицу страны, национальную денежную валюту и континент или острова, где она находится.

9. В двоичной последовательности определить участок одинаковых цифр максимальной длины и выдать: цифру, составляющую это участок, номер позиции начала участка и длину (количество символов) участка (например, $1011010001 \rightarrow$ цифра = 0, позиция = 7, длина = 3).
10. Определить величину вклада в банке на конец года, если на начало года вклад имел величину S , а ежемесячно начисляемый процент равен p .
11. Выдать в возрастающем порядке все трехзначные числа, в десятичной записи которых нет одинаковых цифр.
12. Составить алгоритм перевода числа m , заданного в k -ичной (k от 2 до 9) системе счисления, в число n в десятичной системе счисления.
13. Для заданных чисел x_1, x_2, x_3, x_4, x_5 вычислить $y=x_1\cdot(x_1+x_2)\cdot(x_1+x_2+x_3)\cdot(x_1+x_2+x_3+x_4)\cdot(x_1+x_2+x_3+x_4+x_5)$.
14. Даны две упорядоченные по возрастанию последовательности чисел X и Y размером m и n элементов соответственно. Получить из них упорядоченную по убыванию последовательность Z размером $m+n$ элементов.
15. В матрице размером $m \times n$ поменять местами i и j столбцы.
16. В заданном пятизначном числе определить, есть ли одинаковые цифры и если есть, то какие и их количество (например, $53513 \rightarrow$ “5” две, “3” две).
17. Число, заданное в k -ичной системе счисления, выдать в виде числа, представленного в d -ичной системе счисления (например, $623_8 \rightarrow 447_9$).
18. Для заданной даты дня 2009 года выдать день недели и порядковый номер дня в году.
19. Даны две последовательности чисел A и B . Получить последовательность C , состоящую из элементов A , которых нет в последовательности B .
20. Даны две последовательности чисел A и B . Вычислить сумму и произведение элементов, содержащихся как в A , так и в B .
21. Из матрицы разных чисел A размером $m \times n$ получить матрицу B размером $(m-1) \times (n-1)$ путем вычеркивания строки с минимальным элементом и столбца с максимальным элементом.
22. Написать алгоритм, порождающий все перестановки элементов множества из n чисел, при $n \leq 10$.
23. Выдать задаваемое число в виде десятичного, двоичного, пятеричного и римского чисел. Например: $77_{10} = 1001101_2 = 302_5 = \text{LXXVII}_R$
24. Заполнить и вывести на дисплей квадратную матрицу размера $n \times n$, при четном значении n ($4 \leq n \leq 10$), вида:

1	1	2	2
1	-1	-2	2
4	-3	-4	3
4	4	3	3

25. Определить представляется ли заданное число n в виде суммы квадратов двух натуральных чисел $x^2 + y^2$ двумя различными способами (например, $85 = 7^2 + 6^2 = 9^2 + 2^2$).
26. По заданной строке из n символов сформировать квадратную матрицу размера $n \times n$, расположив символы строки в главной диагонали и заполнив диагонали, параллельные побочной диагонали, соответствующими символами главной диагонали. Остальные элементы матрицы заполнить нулями. Например, для строки “Проба” получаем матрицу

П	0	р	0	о
0	р	0	о	0
р	0	о	0	б
0	о	0	б	0
о	0	б	0	а

27. Треугольником Паскаля называется числовой треугольник, в котором по краям стоят единицы, а каждое число внутри равно сумме двух стоящих над ним в ближайшей строке сверху. Выдать первые n строк треугольника Паскаля.

1				
1	1	1		
1	1	2	1	
1	3	3	1	
1	4	6	4	1
.....				

Пример теста

1. Тело любого цикла выполняется до тех пор, пока его условие ...
 - истинно
 - у цикла нет условия
 - должно
2. Укажите объектно-ориентированный язык программирования
 - Все варианты ответов
 - Eiffel
 - Java
 - C++
3. Какой из следующих операторов - оператор сравнения двух переменных?
 - equal
 - :=
 - =
 - ==
4. Цикл с предусловием?
 - for
 - dowhile
 - while
5. Укажите правильное определение функции main в соответствии со спецификацией стандарта ANSI


```
void main(void)
int main()
int main(void)
void main()
```
6. Какая из следующих записей - правильный комментарий в C++?
 - {комментарий}

- ** Комментарий **
 - /* комментарий */
 - */ Комментарии */
7. Какой из ниже перечисленных операторов, не является циклом в C++?
- for
 - repeat until
 - while
 - dowhile
8. Выберите правильный вариант объявления константной переменной в C++, где type - тип данных в C++ variable - имя переменной value - константное значение
- const type variable = value;
 - const type variable := value;
 - const variable = value;
9. Какой из перечисленных типов данных не является типом данных в C++?
- int
 - real
 - float
 - double
10. Структура объявления переменных в C++
- [=];<идент. 2>,...;
 - [=], <идент. 2>,...;
 - [:=], <идент. 2>,...;
 - [==];<идент. 2>,...;
11. Язык программирования C++ разработал
- Никлаус Вирт
 - Бьорн Страуструп
 - Дональд Кнут
 - Кен Томпсон
12. До каких пор будут выполняться операторы в теле цикла while ($x < 100$)?
- Пока x меньше или равен стам
 - Пока x равен стам
 - Пока x больше ста
 - Пока x строго меньше ста
13. Какое значение, по умолчанию, возвращает программа операционной системе в случае успешного завершения?
- -1
 - 1
 - Программа не возвращает значение.
 - 0
14. Цикл с постусловием?
- while
 - dowhile
 - for
15. Какому зарезервированному слову программа передаёт управление в случае, если значение переменной или выражения оператора switch не совпадает ни с одним константным выражением?
- contingency
 - all
 - other
 - default
16. Что будет напечатано?
- 1 intmain()

```

2 {
3   for (int i = 0; i < 4; ++i)
4   {
5     switch (i)
6     {
7       case 0 : std::cout<< "0";
8       case 1 : std::cout<< "1"; continue;
9       case 2 : std::cout<< "2"; break;
10      default : std::cout<< "D"; break;
11    }
12   std::cout<< ".";
13 }
14 return 0;
15 }
```

- 0.1.2.
- 01.2.D.
- Ошибка компиляции в строке 10
- 011.2.D
- 0112.D.

18. Какие служебные символы используются для обозначения начала и конца блока кода?

- beginend
- ◊◊
- { }
- ()

19. Какой служебный знак ставится после оператора case ?

- .
- -
- :
- ;

20. Чему будет равна переменная a, после выполнения этого кода int a; for(a = 0; a < 10; a++) {}?

- 10
- 1
- 9

21. Каков результат работы следующего фрагмента кода?

```

1 int x = 0;
2
3 switch(x)
4 {
5
6   case 1: cout<< "Один";
7
8   case 0: cout<< "Нуль";
9
10  case 2: cout<< "Привет мир";
11
12 }
```

- Нуль
- НульПривет мир
- Привет мир
- Один

22. Название C++ предложил

- Бьорн Страуструп
- РикМасситти
- Дональд Кнут
- Кэн Томпсон

23. Укажите правильную форму записи цикла dowhile

- ```

1// форма записи оператора цикла dowhile:
2do // начало цикла do while
3{
4/*блок операторов*/;
5}
6while /*условие выполнения цикла*/ // конец цикла dowhile

```
- ```

1// форма записи оператора цикла dowhile:
2do // начало цикла do while
3{
4/*блок операторов*/;
5}
6while /*условие выполнения цикла*/); // конец цикла dowhile
    
```
- ```

1// форма записи оператора цикла dowhile:
2do // начало цикла do while
3{
4/*блок операторов*/;
5}
6while {/*условие выполнения цикла*/} // конец цикла dowhile

```

24. Какими знаками заканчивается большинство строк кода в Си++?

- ; (точка с запятой)
- , (запятая)
- : (двоеточие)
- . (точка)

25. Общий формат оператора множественного выбора - switch

- ```

1switch (switch_expression)
2{
3 case constant1, case constant2: statement1; [break;]
4 case const antN: statement N; [break;]
5 [default: statement N+l;]
6}
    
```
- ```

1switch (switch_expression)
2{
3 case const ant1: statement1; [break;]
4 case const ant2: statement2; [break;]
5 case const antN: statementN; [break;]
6 [else: statementN+l;]
7}

```
- ```

1switch (switch_expression)
2{
3 case const ant1: statement1; [break;]
4 case const ant2: statement2; [break;]
    
```

```

5 caseconst antN: statementN; [break;]
6 [default: statementN+l;]
7}

```

26. Какую функцию должны содержать все программы на C++?

- main()
- system()
- start()
- program()

27. Чтобы подключить заголовочный файл в программу на C++, например, iostream необходимо написать:

- include <iostreamh>
- include #iostream,h;
- #include<>; с iostream.h внутри скобок
- #include<> с iostream внутри скобок

28. Простые типы данных в C++.

- целые – int, вещественные – float или real, символьные – char
- целые – bool, вещественные – float или double, символьные – string
- целые – int, вещественные – float или double, символьные – char
- целые – int, вещественные – float или double, символьные – string

29. Какой оператор не допускает перехода от одного константного выражения к другому?

- break;
- Stop;
- точка с запятой
- end;

30. Программа, переводящая входную программу на исходном языке в эквивалентную ей выходную программу на результирующем языке, называется:

- компилятор
- транслятор
- сканер
- интерпретатор

31. В приведённом коде измените или добавьте один символ чтобы код напечатал 20 звёздочек - *.

```

1inti, N = 20;
2for(i = 0; i < N; i--)
3 printf("*");
•
1int i, N = 40;
2for(i = 0; i < N; i--)
3 printf("*");
•
1int i, N = 20;
2for(i = 19; i < N; i--)
3 printf("*");
•
1int i, N = 20;
2for(i = 0; i < N; N--)
3 printf("*");
•
1int i, N = 20;
2for(i = 20; i < N; i--)
3 printf("*");

```

Вопросы к экзамену

1. Языки программирования. Классификация языков программирования. Этапы программирования (процедурное, модульное, объектно-ориентированное программирование).
2. Этапы создания программного обеспечения. Процесс компиляции программы. Системы программирования. Общая схема работы транслятора.
3. Понятие алгоритма. Свойства алгоритма. Способы записи алгоритма. Основные алгоритмические структуры. Структурный подход к разработке алгоритмов.
4. Язык программирования C++. Структура программы языка C++. Директивы препроцессора `#include`, `#define`.
5. Элементы языка C++: алфавит, комментарии, идентификаторы, служебные слова, литералы, Escape последовательности, операторы, знаки пунктуации.
6. Понятие типов данных. Стандартные типы данных и операции с ними. Выражения. Логические операции.
7. Константы и переменные. Именованные константы, перечисляемый тип.
8. Стандартные функции. Выражения. Правила вычисления значения выражения и определения типа результата. Оператор присваивания. Приведение типов.
9. Программирование последовательности действий на языке Си. Операторы ввода, вывода информации. Формат вывода. Потоковый ввод – вывод.
10. Управляющие структуры. Операция условие. Программирование ветвлений. Условный оператор. Составной оператор. Логические операции и логические выражения.
11. Управляющие структуры. Вложенные условные операторы. Многозначное ветвление. Оператор выбора.
12. Управляющие структуры. Программирование циклов. Цикл с предусловием, с постусловием. Связь между циклами. Простые примеры.
13. Управляющие структуры. Программирование циклов. Цикл с параметром. Различные варианты записи циклов с параметром.
14. Управляющие структуры. Программирование циклов. Вложенные циклы. Операторы `break`, `continue`.
15. Тип массив. Объявление массивов, размещение массивов в памяти. Инициализация элементов массива. Типы задач.
16. Тип массив. Сортировка массива по возрастанию: линейная сортировка (отбором) и пузырьковая сортировка.
17. Тип массив. Объявление многомерных массивов, размещение многомерных массивов в памяти. Передача массивов в функции.
18. Строковый тип. Особенности представления строковых данных в C++. Стандартные процедуры и функции для обработки строковых величин.
19. Указатели. Операции с указателями. Ссылочный тип данных.
20. Указатели и массивы.
21. Функции. Объявление функции. Оператор `return`. Прототипы функций.
22. Функции. Передача значений в функцию. Классы памяти.
23. Функции. Перегруженные имена функций. Аргументы по умолчанию.
24. Функции. Рекурсия.
25. Структуры и объединения. Операторы доступа.
26. Файлы данных. Ввод-вывод в файл. Обработка файлов данных.
27. Объектно-ориентированный подход. Основные понятия и определения. Определение класса. Уровни видимости.
28. Объектно-ориентированный подход. Методы класса. Конструкторы и деструктор.

Примеры экзаменационных задач

1. Даны две действительные квадратные матрицы порядка n . Получить новую матрицу умножением элементов каждой строки первой матрицы на наибольший из значений элементов соответствующей строки второй матрицы.
2. Для заданного натурального числа найти наибольшее, составленное из тех же цифр.
3. Задан текст. Подсчитать сколько различных букв использовано при написании этого текста. (Текст: мама. Результат работы программы – 2 буквы.)
4. Даны две строки А и В. Напишите программу, которая определяет, можно ли из символов, входящих в строку А, составить строку В. Символы можно переставлять, но использовать каждый из них разрешается не более одного раза. (А – программист, В – программа. Результат работы программы – нельзя.)
5. Даны целые числа a_1, a_2, \dots, a_n . Найти среди них 2 числа, модуль разности которых имеет наибольшее значение.

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 БГПУ;
- система электронного обучения ФГБОУ ВО «БГПУ»;
- система тестирования на основе единого портала «Интернет-тестирования в сфере образования www.i-exam.ru»;
- электронные библиотечные системы;
- мультимедийное сопровождение лекций и практических занятий;
- система программирования Dev-C++ или подобные ей.

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 C++: 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 C++: 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, C++, VisualBasic [Текст]: учеб.-справочное пособие / Ю. А. Аляев, О. А. Козлов. - М.: Финансы и статистика, 2007. - 318, [1] с. (11 экз.)
4. Гниденко И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. – Москва: Издательство Юрайт, 2022. – 235 с. – (Высшее образование). – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/viewer/tehnologii-i-metody-programmirovaniya-489920> (дата обращения: 06.09.2022).
5. 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.)
6. Иванова, Г. С. Технология программирования : учебник для студ. вузов / Г. С. Иванова. - М. : КНОРУС, 2011. - 333 с. (6 экз.)
7. Ишкова, Э.А. C++ начала программирования: учебник / Ишкова Э.А. – 3-е изд., испр.и доп. – М.: ООО «Бином-Пресс», 2004. – 368 с. (17 экз.)
8. Павловская, Т.А. С/C++ 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.).
9. Павловская, Т.А. С/C++ 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.)
10. С/C++. Алгоритмы и приемы программирования / Тимофеева В.,ред. – М.: Бином, 2003. – 560 с. (6 экз.)
11. Сальников, Ю. Н. Программирование. Базовый курс : учеб. пособие / Ю.Н. Сальников. – М. : Маркет DC, 2010. - 335 с. (14 экз.)
12. Огнева, М. В. Программирование на языке C++: практический курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. – Москва : Издательство Юрайт, 2022. – 335 с. – (Высшее образование). – ISBN 978-5-534-05123-0. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/492984> (дата обращения: 05.10.2022).

9.2 Базы данных и информационно-справочные системы

1. Федеральный портал «Российское образование» – Режим доступа: <http://www.edu.ru>
2. Информационная система «Единое окно доступа к образовательным ресурсам» – Режим доступа: <http://www.window.edu.ru>
3. Федеральный центр информационно-образовательных ресурсов – Режим доступа: <http://fcior.edu.ru>
4. 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.
- Стол преподавателя.
- Аудиторная доска.
- Компьютеры с установленным лицензионным специализированным программным обеспечением.
- Мультимедийный проектор.
- Экспозиционный экран.
- Учебно-наглядные пособия – мультимедийные презентации по дисциплине «Программирование».

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; 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 2025/2026 уч. г.

РПД обсуждена и одобрена для реализации в 2025/2026 уч. г. без изменений на заседании кафедры информатики и методики преподавания информатики (протокол №6 от 26.05.2025 г.).