



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**  
**Б1.О.04.11 Языки и методы программирования**

**Направление подготовки:** 01.03.02 Прикладная математика и информатика

**Направленность (профиль):** Компьютерное моделирование экономических процессов

**Квалификация (степень):** бакалавр

**Форма обучения:** очная

**Институт:** математики, естествознания и техники

**Кафедра:** математического моделирования и компьютерных технологий

	очная форма	очно- заочная форма	заочная форма
Курс	1-2		
Семестр	2, 3		

Лекции	36		
Лабораторные занятия	18		
Практические (семинарские) занятия	18		
Консультации	2		
Форма(ы) промежуточной аттестации	Экзамен - 0,3 Зачет с оценкой – 0,2		
Контроль	36		
Иные формы работы	–		
Самостоятельная работа	105,5		

**Всего часов: 216**

**Трудоемкость: 6 зачетных единиц.**

Разработчик рабочей программы:

кандидат физико-математических наук, доцент Д.В. Корниенко

## I. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ РАЗДЕЛ

### Цель изучения дисциплины:

- овладение знаниями и навыками программирования с использованием современных языков программирования;
- формирование компьютерной грамотности и подготовка студентов к использованию современных компьютеров и технологий программирования в качестве инструмента для решения практических задач в своей предметной области;
- формированием представлений о роли программирования в управлении и специфике информационных систем;
- формированием информационной культуры, подготовка будущих бакалавров к практической деятельности в условиях широкого использования информационных технологий.

### Задачи изучения дисциплины:

- овладение приемами работы в современных средах разработки программного обеспечения, обеспечивающих широкие возможности написания, редактирования и компилирования программ;
- обучение принципам построения программного кода;
- формирование навыков программирования на языке высокого уровня, которые будут использоваться и углубляться при выполнении различных заданий и работ по дисциплинам, изучаемым на последующих курсах;
- развитие навыков системного мышления;
- систематизация и расширение знаний приемов и методов работы с информационно-коммуникационными технологиями, подготовка к их осознанному использованию при решении различного вида прикладных задач.

Место дисциплины в структуре ОПОП: реализуется в рамках обязательной части блока Б1. Дисциплины (модули).

### Планируемые результаты обучения по дисциплине:

Код компетенции	Индикаторы достижения компетенции	Планируемые результаты обучения по дисциплине
ОПК-2	<b>Знать:</b> - основы математического моделирования и системного программирования;	<b>Знает:</b> – методологии разработки программного обеспечения и технологии программирования; – языки формализации функциональных спецификаций – методы и приемы формализации задач; – методы и средства проектирования программного обеспечения, программных интерфейсов и баз данных; – принципы построения и виды архитектуры программного обеспечения;

		– типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения;
	<b>Уметь:</b> - применять математическое и программное обеспечение, прикладные интернет-технологии, автоматизированные системы, средства компьютерной графики к решению прикладных задач;	<b>Умеет:</b> – использовать существующие типовые решения и шаблоны проектирования программного обеспечения; – применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов; – осуществлять коммуникации с заинтересованными сторонами; – выбирать средства реализации требований к программному обеспечению; – вырабатывать варианты реализации программного обеспечения и требований к нему; – проводить анализ исполнения требований
	<b>Владеть:</b> - действиями по математическому и программному обеспечению процессов решения прикладных задач в сфере управления предприятием, в сфере сетевых технологий, баз данных.	<b>Владеет</b> – методологией и технологиями проектирования программного обеспечения, программных интерфейсов, структур и баз данных в соответствии с установленными требованиями; – действиями по разработке и согласованию технических спецификаций на программные компоненты; – действиями по согласованию требований к программному обеспечению с заинтересованными сторонами, распределению заданий между программистами в соответствии с техническими спецификациями, осуществлению контроля выполнения заданий, формированию отчетности в соответствии с установленными регламентами

## II. СОДЕРЖАНИЕ И ОБЪЕМ ДИСЦИПЛИНЫ

с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу

### Очная форма обучения

№ п/п	Наименование разделов и тем	Всего	Аудиторные занятия			Сам. раб.
			ЛК	ПЗ	ЛБ	
2 семестр						
Раздел 1. Структурированное программирование		105,7	18	18		69,7
1	Тема 1. Структурная разработка программ (Алгоритмы. Структуры выбора. Операции присваивания. Опе- рации инкремента и декремента)	8	1	1		6
2	Тема 2.	10	2	2		6

	Управление программой (Основы структур повторения. Структура со множественным выбором)					
3	<b>Тема 3.</b> Функции (Программные модули. Функции математической библиотеки. Определения функций. Прототипы функций. Вызов функций: вызов по значению и по ссылке. Классы памяти. Правила области действия. Рекурсия)	10	2	2		6
4	<b>Тема 4.</b> Массивы (Объявление массивов. Примеры работы с массивами. Передача массивов в функции. Сортировка массивов. Поиск в массивах. Многомерные массивы)	20	4	4		12
5	<b>Тема 5.</b> Указатели (Объявления и инициализация переменной-указателя. Операции над указателями. Передача параметра по ссылке. Связь между указателями и массивами. Массивы указателей. Указатели на функции)	14	2	2		10
6	<b>Тема 6.</b> Символы и строки (Библиотека обработки символов. Функции преобразования строк. Функции стандартной библиотеки ввода/вывода)	10	1	1		8
7	<b>Тема 7.</b> Структуры, объединения и перечисления (Описания структур. Инициализация структур. Доступ к элементам структур. Использование структур с функциями. Объединения. Структуры, ссылающиеся на себя. Динамическое распределение памяти. Связанные списки. Стеки. Очереди. Деревья)	20	4	4		12
8	<b>Тема 8.</b> Работа с файлами (Файлы и потоки. Создание файла последовательного доступа. Чтение данных из файла последовательного доступа. Файлы произвольного доступа. Создание файла произвольного доступа. Произвольная запись данных в файл произвольного доступа. Последовательное чтение данных из файла произвольного доступа)	13,7	2	2		9,7
	<i>Контроль</i>	36				
	<i>Консультации</i>	2				
	<i>Экзамен</i>	0,3				
	<i>Итого за 2 семестр</i>	144	18	18		69,7
<b>3 семестр</b>						
<b>Раздел 2.</b> <b>Объектно-ориентированное программирование</b>		<b>71,8</b>	<b>18</b>		<b>18</b>	<b>35,8</b>
9	<b>Тема 9.</b> Понятия объектно-ориентированного программирования	8,8	2		2	4,8

10	<b>Тема 10.</b> Классы и абстракция данных (Создание класса. Область действия класса и доступ к элементам класса. Функции доступа и сервисные функции. Инициализация объектов класса: конструкторы. Использование с конструкторами аргументов по умолчанию. Деструкторы. Использование элементов данных и элементов-функций)	16	4		4	8
11	<b>Тема 11.</b> Перегрузка операций (Основные принципы перегрузки операций. Запреты на перегрузку операций)	8	2		2	4
12	<b>Тема 12.</b> Наследование (Базовые и производные классы. Защищенные элементы. Переопределение элементов базового класса в производном классе. Открытые, защищенные и закрытые базовые классы. Непосредственные и косвенные базовые классы)	18	4		4	10
13	<b>Тема 13.</b> Виртуальные функции и полиморфизм (Виртуальные функции. Абстрактные базовые классы и конкретные классы. Полиморфизм. Новые классы и динамическое связывание)	21	6		6	9
	<i>Зачет с оценкой</i>	0,2				
	<i>Итого за 3 семестр</i>	72	18		18	35,8
	<b>ИТОГО:</b>	<b>216</b>	<b>36</b>	<b>18</b>	<b>18</b>	<b>105,5</b>

**Очно-заочная форма обучения не реализуется**

**Заочная форма обучения не реализуется**

### **III. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

Текущая аттестация проводится в форме контрольной работы, теста, реферата.

#### **Типовой вариант контрольной работы**

##### **Контрольная работа №1 (7 вариантов)**

1. Алгоритмы. Структуры выбора. Операции присваивания. Операции инкремента и декремента.
2. Основы структур повторения. Структура со множественным выбором.
3. Программные модули. Функции математической библиотеки. Определения функций. Прототипы функций. Вызов функций: вызов по значению и по ссылке. Классы памяти. Правила области действия. Рекурсия.
4. Объявление массивов. Примеры работы с массивами. Передача массивов в функции. Сортировка массивов. Поиск в массивах. Многомерные массивы.

5. Объявления и инициализация переменной-указателя. Операции над указателями. Передача параметра по ссылке. Связь между указателями и массивами. Массивы указателей. Указатели на функции.
6. Строки и символы. Библиотека обработки символов. Функции преобразования строк. Функции стандартной библиотеки ввода/вывода.
7. Форматированный вывод. Форматированный ввод.

### Контрольная работа №2 (5 вариантов)

1. Класс и объект. Назначение элементов и методов класса.
2. Конструктор и деструктор класса.
3. Дружественные функции и классы.
4. Наследование класса. Пример простого наследования класса.
5. Множественное наследование. Базовый и производный классы.

### Типовой вариант тестов (1 вариант)

**A1.** Определите значение каждой из переменных после выполнения вычисления  $y *= x++$ . Предположите, что перед началом выполнения оператора все переменные равны 5.

- a)  $x = 6, y = 25$
- b)  $x = 5, y = 25$
- c)  $x = 5, y = 36$
- d)  $x = 6, y = 36$

**A2.** Что будет выведено на экран в результате выполнения программы:

```
#include <iostream>
void func (int num);
int main(void)
{
    func(5);
    return 0;
}
void func(int num)
{
    if (num)
        func(num-1);
    std::cout << num;
}
```

- a) 012345;
- b) программа не скомпилируется;
- c) программа приведет к переполнению стека;
- d) 543210.

**A3.** Что напечатает следующий код:

```
#include "stdio.h"
int main()
{
    int * param1 = new int(2);
    int param2 = 2;
    int * reference = &param2;
```

```

        printf("%d", *reference == *param1);
        return 0;
    }

```

- a) true;
- b) false;
- c) 0;
- d) 1;
- e) возникнет ошибка компиляции

**A4.** Какие значения будет иметь массив msv:

```

int main()
{
    int msv[10];
    for (int i = 0; i < 10; i++)
        msv[i] = i;
}

```

- a) 1,2,3,4,5,6,7,8,9,10;
- b) 0,1,2,3,4,5,6,7,8,9;
- c) Произойдет ошибка времени выполнения;
- d) Программа не скомпилируется.

**A5.** Что выведется на экран после выполнения данной программы?

```

void function (int *);
int main()
{
    int a=1;
    void function (&a);
    printf("%d", a);
}

void function (int *b)
{
    *b=*b+3;
}

```

- a) 3
- b) 1
- c) 4
- d) Произойдет ошибка времени выполнения

**A6.** Отметьте строки с ошибкой инициализации:

```

struct A
{
    int i;
    static int j;
    int k;
};

```

- a) A a = { };
- b) A b = { 1 };

- c) A c = {1,2};
- d) A d = {1,2,3};

**A7.** Как можно вызвать метод следующего класса:

```
class Child
{
public:
    void Count(){ }
};
```

- a) Child \* obj = new Child;  
obj->Count;
- b) Child obj = new Child;  
obj.Count();
- c) Child \* obj = new Child;  
obj.Count();
- d) Child obj = new Child;  
(\*obj).Count();

**A8.** Допустимо ли в C++ определение следующего чисто виртуального метода:

```
class Abstract
{
public:
    virtual voidPure ()= 0
    {
    }
};
```

- a) Да, метод доступен по имени PureVirtual();
- b) Да, метод доступен по имени Abstract: PureVirtual();
- c) Да, однако метод не доступен для вызова;
- d) Нет, определение не допустимо.

**A9.** Что не верно в следующем коде:

```
class Parent
{
public:
    ~Parent(){ }
    virtual void method() { }
};

class Child : public Parent
{
public:
    Child() { /*Захват ресурсов*/ }
    ~Child() { /*Освобождение ресурсов*/ }
    void method() { /*программный код*/ }
};

main()
{
```



```

        Parent *obj=new Child;
        delete obj;
        return 0;
    }

```

- a) Код полностью корректен;
- b) Деструктор базового класса необходимо объявить как virtual;
- c) Код по очистке ресурсов необходимо перенести в базовый класс;
- d) Метод method в базовом классе не нужно объявлять как virtual;
- e) Деструктор и метод класса- наследника необходимо объявить как virtual.

**A10.** Что напечатает код при создании экземпляра класса X:

```

class Y
{
public:
    Y() {cout<<"Y";}
};

class Z
{
public:
    Z() {cout<<"Z";}
};

class X:public Z
{
private:
    Y m_objY;
public:
    X() {cout<<"X";}
};

```

- a) XY;
- b) XYZ;
- c) YXZ;
- d) ZYX.

### Примерная тематика рефератов

1. Распределенные и параллельные методы программирования
2. Современные методы Web-ориентированного программирования на языке C#
3. Web-технологии в разработке удаленных баз данных
4. Применение case-технологий к проектированию базы данных
5. Программные методы коррекции изображения
6. Способы обработки изображений с использованием библиотеки компьютерного зрения OpenCV
7. Построение многомодульных приложений
8. Разработка Web-приложений с применением технологии ASP.NET
9. Разработка клиент-серверного приложения средствами языка C#

Промежуточная аттестация обучающихся осуществляется в форме экзамена во 2 семестре и зачета с оценкой в 3 семестре с использованием следующих оценочных материалов:

### **Вопросы к экзамену (2 семестр, очная форма обучения)**

1. Структура повторения while. Структура повторения for. Структура со множественным выбором switch
2. Операции инкремента и декремента
3. Операторы break и continue
4. Функции математической библиотеки. Рекурсия
5. Классы памяти. Правила области действия
6. Объявление массивов. Передача массивов в функции. Сортировка массивов.
7. Многомерные массивы
8. Объявление и инициализация переменной-указателя. Операции над указателями. Связь между указателями и массивами. Массивы указателей. Указатели на функции
9. Описание структур. Объединения. Структуры, ссылающиеся на себя
10. Динамическое распределение памяти
11. Связанные списки

### **Задачи к экзамену (2 семестр, очная форма обучения)**

**1** Напишите программу на C, которая моделирует бросание двух игральных костей. Программа должна вызывать функцию rand для «бросания» первой кости и затем эту же функцию для «бросания» второй. Затем должна быть вычислена сумма двух значений. Замечание: поскольку на каждой кости может выпасть целое значение от 1 до 6, то сумма двух значений может изменяться от 2 до 12, при этом 7 будет наиболее часто встречающейся суммой, а 2 и 12 -встречающимися наименее часто. Ваша программа должна бросить две кости 36 раз. Используйте одномерный массив для подсчета числа появлений каждой из возможных сумм. Выведите результаты в табличной форме. Также определите, являются ли итоговые суммы правдоподобными, т.е. поскольку для выбрасывания 7 существует шесть способов, то все выбрасывания 7 должны составить приблизительно одну шестую.

**2** (*Система предварительной продажи билетов*). Небольшая авиакомпания недавно приобрела компьютер для новой автоматизированной системы предварительной продажи билетов. Президент компании попросил вас написать на C программное обеспечение для новой системы. Вы должны составить программу для бронирования мест на каждом рейсе единственного самолета авиакомпании (его вместимость: 10 мест). Ваша программа должна отображать на экране следующее меню альтернативных возможностей:

Пожалуйста, введите 1 для «курящих»

Пожалуйста, введите 2 для «некурящих»

Если оператор вводит 1, то ваша программа должна забронировать место в отсеке для курящих (места 1-5). Если оператор вводит 2, то ваша программа должна забронировать место в отсеке для некурящих (места 6-10). После этого ваша программа должна напечатать посадочный талон с указанием номера места пассажира и информации о том, находится ли оно в отсеке для курящих или в отсеке для некурящих пассажиров самолета. Для представления схемы мест для пассажиров самолета используйте одномерный массив. Инициализируйте все элементы массива нулями, чтобы показать, что все места свободны. По мере бронирования каждого места устанавливайте соответствующие элементы массива в 1 для указания на то, что это место больше не является свободным. Конечно, ваша программа никогда не должна бронировать уже забронированное место. Если отсек для курящих заполнен, ваша программа должна запросить оператора, допустимо ли бронирование места в отсеке для некурящих (и наоборот). В случае положительного ответа произведите соответствующее бронирование места. В случае отрицательного ответа выведите сообщение «Следующий рейс через 3 часа.»

**3** Используйте двумерный массив для решения следующей задачи. В компании работают четыре продавца (с 1 по 4), которые продают пять различных видов изделий (с 1 по 5). Один раз в день каждый продавец передает в компанию карточку сбыта по каждой разновидности проданного товара. Каждая карточка содержит:

1. Номер продавца
2. Номер изделия
3. Общую сумму в долларах за данный товар, проданный в этот день

Таким образом, каждый продавец передает в день от 0 до 5 карточек сбыта. Предположим, что доступна информация по всем карточкам за последний месяц. Напишите программу, которая будет считывать всю эту информацию о сбыте за последний месяц и подводить общий итог о сбыте каждым продавцом каждой разновидности товара. Все итоговые суммы должны храниться в двумерном массиве `sales`. После обработки всей информации за последний месяц выведите результаты в виде таблицы, в которой каждый столбец представляет конкретного продавца и каждая строка представляет конкретную разновидность товара. Для получения общего сбыта каждой разновидности товара за последний месяц просуммируйте каждую строку; для получения общего сбыта для каждого продавца за последний месяц проведите суммирование каждого столбца. Распечатка вашей таблицы должна включать эти перекрестные итоговые суммы справа от итоговых строк и в нижней части итоговых столбцов.

**4** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) сумму отрицательных элементов массива;
- 2) произведение элементов массива, расположенных между максимальным и минимальным элементами.

Упорядочить элементы массива по возрастанию.

**5** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) сумму положительных элементов массива;

2) произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.

Упорядочить элементы массива по убыванию.

**6** В одномерном массиве, состоящем из  $n$  целых элементов, вычислить:

- 1) произведение элементов массива с четными номерами;
- 2) сумму элементов массива, расположенных между первым и последним нулевыми элементами.

Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом - все отрицательные (элементы, равные 0, считать положительными).

**7** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) сумму элементов массива с нечетными номерами;
- 2) сумму элементов массива, расположенных между первым и последним отрицательными элементами.

Сжать массив, удалив из него все элементы, модуль которых не превышает 1. Освободившиеся в конце массива элементы заполнить нулями.

**8** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) максимальный элемент массива;
- 2) сумму элементов массива, расположенных до последнего положительного элемента.

Сжать массив, удалив из него все элементы, модуль которых находится в интервале  $[a, b]$ . Освободившиеся в конце массива элементы заполнить нулями.

**9** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) минимальный элемент массива;
- 2) сумму элементов массива, расположенных между первым и последним положительными элементами.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, равные нулю, а потом — все остальные.

**10** В одномерном массиве, состоящем из  $n$  целых элементов, вычислить:

- 1) номер максимального элемента массива;
- 2) произведение элементов массива, расположенные между первым и вторым нулевыми элементами.

Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй половине - элементы, стоявшие в четных позициях.

**11** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) номер минимального элемента массива;

2) сумму элементов массива, расположенных между первым и вторым отрицательными элементами.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, модуль которых не превышает 1, а потом — все остальные.

**12** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) максимальный по модулю элемент массива;
- 2) сумму элементов массива, расположенных между первым и вторым положительными элементами.

Преобразовать массив таким образом, чтобы элементы, равные нулю, располагались после всех остальных.

**13** В одномерном массиве, состоящем из  $n$  целых элементов, вычислить:

- 1) минимальный по модулю элемент массива;
- 2) сумму модулей элементов массива, расположенных после первого элемента, равного нулю.

Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в четных позициях, а во второй половине — элементы, стоявшие в нечетных позициях.

**14** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) номер минимального по модулю элемента массива;
- 2) сумму модулей элементов массива, расположенных после первого отрицательного элемента.

Сжать массив, удалив из него все элементы, величина которых находится в интервале  $[a, b]$ . Освободившиеся в конце массива элементы заполнить нулями.

**15** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) количество элементов массива, лежащих в диапазоне от  $A$  до  $B$ ;
- 2) сумму элементов массива, расположенных после максимального элемента. Упорядочить элементы массива по убыванию модулей элементов.

**16** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) количество элементов массива, равных 0;
- 2) сумму элементов массива, расположенных после минимального элемента. Упорядочить элементы массива по возрастанию модулей элементов.

**17** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) количество элементов массива, больших  $C$
- 2) произведение элементов массива, расположенных после максимального по модулю элемента.

Преобразовать массив таким образом, чтобы сначала располагались все отрицательные элементы, а потом — все положительные (элементы, равные 0, считать положительными).

**18** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) количество отрицательных элементов массива;
- 2) сумму модулей элементов массива, расположенных после минимального по модулю элемента.

Заменить все отрицательные элементы массива их квадратами и упорядочить элементы массива по возрастанию.

**19** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) количество положительных элементов массива;
- 2) сумму элементов массива, расположенных после последнего элемента, равного нулю.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых не превышает 1, а потом — все остальные.

**20** В одномерном массиве, состоящем из  $n$  вещественных элементов, вычислить:

- 1) произведение положительных элементов массива;
- 2) сумму элементов массива, расположенных до минимального элемента. Упорядочить по возрастанию отдельно элементы, стоящие на четных местах, и элементы, стоящие на нечетных местах.

**21** Дана целочисленная прямоугольная матрица. Определить:

- 1) количество строк, не содержащих ни одного нулевого элемента;
- 2) максимальное из чисел, встречающихся в заданной матрице более одного раза.

**22** Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного нулевого элемента.

Характеристикой строки целочисленной матрицы назовем сумму ее положительных четных элементов. Переставляя строки заданной матрицы, расположить их в соответствии с ростом характеристик.

**23** Дана целочисленная прямоугольная матрица. Определить:

- 1) количество столбцов, содержащих хотя бы один нулевой элемент;
- 2) номер строки, в которой находится самая длинная серия одинаковых элементов.

**24** Дана целочисленная квадратная матрица. Определить:

- 1) произведение элементов в тех строках, которые не содержат отрицательных элементов;
- 2) максимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы.

**25** Дана целочисленная квадратная матрица. Определить:

- 1) сумму элементов в тех столбцах, которые не содержат отрицательных элементов;

2) минимум среди сумм модулей элементов диагоналей, параллельных побочной диагонали матрицы.

**26** Для заданной матрицы размером 8 на 8 найти такие  $k$ , что  $k$ -я строка матрицы совпадает с  $k$ -и столбцом.

Найти сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент.

**27** Характеристикой столбца целочисленной матрицы назовем сумму модулей его отрицательных нечетных элементов. Переставляя столбцы заданной матрицы, расположить их в соответствии с ростом характеристик.

Найти сумму элементов в тех столбцах, которые содержат хотя бы один отрицательный элемент.

**28** Элемент матрицы называется локальным минимумом, если он строго меньше всех имеющихся у него соседей. Подсчитать количество локальных минимумов заданной матрицы размером 10 на 10.

Найти сумму модулей элементов, расположенных выше главной диагонали.

**29** Упорядочить строки целочисленной прямоугольной матрицы по возрастанию количества одинаковых элементов в каждой строке.

Найти номер первого из столбцов, не содержащих ни одного отрицательного элемента.

**30** Дана целочисленная квадратная матрица. Определить:

- 1) сумму элементов в тех строках, которые не содержат отрицательных элементов;
- 2) минимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы.

### **Вопросы к зачету с оценкой (3 семестр, очная форма обучения)**

1. Создание класса. Область действия класса и доступ к элементам класса. Функции доступа и сервисные функции.
2. Инициализация объектов класса: конструкторы. Использование с конструкторами аргументов по умолчанию. Деструкторы.
3. Основные принципы перегрузки операций.
4. Запреты на перегрузку операций.
5. Базовые и производные классы.
6. Переопределение элементов базового класса в производном классе.
7. Виртуальные функции.
8. Полиморфизм.
9. Распределенные и параллельные методы программирования.
10. Построение многомодульных приложений.

11. Стек.
12. Очередь.
13. Дерево.

### **Задачи к зачету с оценкой (3 семестр, очная форма обучения)**

1. Создайте класс TIME (время), который содержит функции позволяющие установить часы, минуты и секунды (SET функции), а также функции возвращающие значения часов, минут и секунд (GET функции). Объекты этого класса должны выводиться в военном (23:45:32) и стандартном (11:45:32) форматах.
2. Создайте класс DATE (дата), который содержит функции позволяющие установить год, месяц и день (SET функции), а также функции возвращающие значения года, месяца и дня (GET функции). Объекты этого класса должны выводить дату.
3. Создайте класс EMPLOYEE (служащий), который содержит в себе объекты класса DATE (дата). Один объект класса DATE символизирует дату рождения служащего, второй объект класса DATE символизирует дату поступления на работу. Объекты класса EMPLOYEE выводят имя, фамилию, дату рождения и дату поступления на работу служащего.
4. Создайте класс PhoneNumber (телефонный номер), закрытые элементы которого есть код города и номер. Также создайте дружественные функции, которые перегружают операции передачи и извлечения из потока и позволяющие обрабатывать данные класса телефонного номера.
5. Создайте класс Point (точка), содержащий координаты точки. Создайте класс Circle (окружность) производный от класса Point, содержащий элемент данных радиус. Программа должна выводить объект класса Circle в виде - Центр = [x; y]; Радиус = r .
6. Создайте класс Shape (форма). Создайте от него производные классы Circle (окружность) и Cylinder(цилиндр). Класс Shape должен содержать чисто виртуальную функцию printShapeName (печатать имя формы). Эта функция должна переопределяться в каждом производном классе. Имена объектов производных классов должны выводиться на печать через указатель на базовый класс, то есть использовать свойства полиморфизма.
7. Создайте класс стек, который вставляет переменную в стек и выталкивает ее.
8. Создайте класс сотовый телефон. Создайте класс прилавок. Причем класс прилавок должен содержать не менее 3 объектов класса сотовый телефон. При описании классов используйте следующую информацию. Класс сотовый телефон должен содержать следующую информацию о сотовом телефоне:
  - 1) наименование фирмы производителя
  - 2) количество часов в режиме разговора
  - 3) количество часов в режиме ожидания
  - 4) число 1 если цветной дисплей и 0 если не цветной дисплей
  - 5) ценаКласс прилавок должен содержать следующую информацию:



- 1) порядковый номер прилавка
- 2) количество сотовых телефонов на прилавке
- 3) информацию об имеющихся телефонах

**10.** Создайте класс группа. Создайте класс факультет. Причем класс факультет должен содержать не менее 3 объектов класса группа. При описании классов используйте следующую информацию. Класс группа должен содержать следующую информацию о группе:

- 1) наименование группы (к примеру, ПМ-11, ПМ-21)
- 2) количество человек в группе
- 3) фамилию старосты

Класс факультет должен содержать следующую информацию:

- 1) название факультета
- 2) количество групп
- 3) информацию о каждой группе

**11.** Создайте класс Student (Студент). Класс должен содержать в закрытых элементах-данных переменную, которая отвечает за фамилию студента, переменную, которая отвечает за имя студента, переменную, которая отвечает за название группы в которой обучается студент и переменную в которой хранится возраст студента. В открытом интерфейсе класса должны присутствовать функция, которая устанавливает значения для студента и функция, которая печатает информацию о студенте.

**12.** Создайте класс Преподаватель. Создайте класс Факультет. Причем класс Факультет должен содержать объекты класса Преподаватель. При описании классов используйте следующую информацию. Класс Преподаватель должен содержать следующую информацию о преподавателе:

- 1) Фамилию
- 2) Имя
- 3) Ученую степень
- 4) Звание
- 5) Должность

Класс Факультет должен содержать следующую информацию о факультете:

- 1) Название факультета
- 2) Декан факультета
- 3) Преподаватели факультета (2 человека).

**13.** Создайте класс Многочлен (3 степени). В классе должны присутствовать следующие функции: инициализация многочлена, вывод на экран многочлена, перегруженная операция «+».

**14.** Создайте класс Комплексное Число. В классе должны присутствовать функции, которые перегружают следующие операции: «+», «-», «\*».

**15.** Создайте класс Матрица и реализуйте в нем функции, отвечающие за сложение матриц, умножения матрицы на число.

## IV. ПЕРЕЧЕНЬ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

### 4.1. Основная литература

1. Гунько, А.В. Программирование : учебно-методическое пособие : [16+] / А.В. Гунько ; Новосибирский государственный технический университет. – Новосибирск : Новосибирский государственный технический университет, 2019. – 74 с. : табл. – Режим доступа: по подписке. – URL: [https://biblioclub.ru/index.php?page=book\\_view\\_red&book\\_id=576267](https://biblioclub.ru/index.php?page=book_view_red&book_id=576267) (дата обращения: 01.09.2020). – Библиогр. в кн. – ISBN 978-5-7782-3961-6. – Текст : электронный.
2. Горелов, С.В. Современные технологии программирования: разработка Windows-приложений на языке C#: учебник для студентов, обучающихся по дисциплине «Современные технологии программирования», направление «Прикладная информатика» (09.03.03 — для бакалавров, 09.04.03 — для магистров) : в 2 томах : [16+] / С.В. Горелов ; под науч. ред. П.Б. Лукьянова ; Финансовый университет при Правительстве Российской Федерации. – Москва : Прометей, 2019. – Том 1. – 363 с. : ил. – Режим доступа: по подписке. – URL: [https://biblioclub.ru/index.php?page=book\\_view\\_red&book\\_id=576037](https://biblioclub.ru/index.php?page=book_view_red&book_id=576037) (дата обращения: 01.09.2020). – Библиогр. в кн. – ISBN 978-5-907100-09-1. – Текст : электронный.

### 4.2. Дополнительная литература

1. Практикум по администрированию программного обеспечения: лабораторный практикум : [16+] / авт.-сост. И.В. Анзин ; Северо-Кавказский федеральный университет. – Ставрополь : Северо-Кавказский Федеральный университет (СКФУ), 2017. – 85 с. : ил. – Режим доступа: по подписке. – URL: [https://biblioclub.ru/index.php?page=book\\_view\\_red&book\\_id=483755](https://biblioclub.ru/index.php?page=book_view_red&book_id=483755) (дата обращения: 01.09.2020). – Библиогр. в кн. – Текст : электронный.

## V. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

№ пп	Ссылка на информационный ресурс	Наименование разработки в электронной форме	Доступность
1.	<a href="http://edu.ru/">http://edu.ru/</a>	Российское образование: Федеральный портал. Включает ссылки на порталы и сайты образовательных учреждений; государственные образовательные стандарты; нор-	Свободный доступ

		мативные документы; каталог экскурсий и обучающих программ.	
2.	<a href="http://citforum.ru/database/osbd/contents.shtml">http://citforum.ru/database/osbd/contents.shtml</a>	Информационно-аналитические материалы	Свободный доступ

## **VI. СОВРЕМЕННЫЕ ПРОФЕССИОНАЛЬНЫЕ БАЗЫ ДАННЫХ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ**

1.	<a href="http://www.biblioclub.ru">http://www.biblioclub.ru</a>	Электронно-библиотечная система (ЭБС) Университетская библиотека онлайн	Регистрация через любой университетский компьютер. В дальнейшем предоставляется неограниченный индивидуальный доступ из любой точки, в которой имеется доступ к сети Интернет
2.	<a href="http://www.garant.ru">www.garant.ru</a>	Информационно-правовой портал	Свободный доступ
3.	<a href="http://www.elibrary.ru">www.elibrary.ru</a>	Российский информационный портал в области науки, технологии, медицины и образования	Свободный доступ
4.	<a href="http://www.consultant.ru">www.consultant.ru</a>	Российская компьютерная справочно-правовая система	Свободный доступ

## **VII. ЛИЦЕНЗИОННОЕ И СВОБОДНО РАСПРОСТРАНЯЕМОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**

При реализации учебной дисциплины применяется следующее лицензионное и свободно распространяемое программное обеспечение:

- Microsoft Windows;
- Microsoft Office;
- LibreOffice;
- и др.

## **VIII. ОБОРУДОВАНИЕ И ТЕХНИЧЕСКИЕ СРЕДСТВА ОБУЧЕНИЯ, НЕОБХОДИМЫЕ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ**

Учебные занятия проводятся в аудиториях, укомплектованных специализированной мебелью, в том числе стационарными или переносными техническими средствами обучения (проектор, экран, компьютер/ноутбук).

Лабораторные занятия, групповые и индивидуальные консультации, текущая и промежуточная аттестации проводятся в специализированных классах, оснащенных автоматизированными рабочими местами с компьютерами.

Самостоятельная работа проводится в кабинетах, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.