

Министерство образования и науки Российской Федерации  
 Федеральное государственное бюджетное образовательное учреждение  
 высшего профессионального образования  
 «Владимирский государственный университет  
 имени Александра Григорьевича и Николая Григорьевича Столетовых»  
 (ВлГУ)



УТВЕРЖДАЮ

Проректор по УМР  
 А.А.Панфилов

« 14 » 04 2015 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**ОСНОВЫ ПРОГРАММИРОВАНИЯ**

(наименование дисциплины)

Направление подготовки: 02.03.03 Математическое обеспечение и администрирование информационных систем

Профиль/программа подготовки:

Уровень высшего образования: бакалавриат

Форма обучения: очно-заочная, ускоренная

Семестр	Трудоемкость зач. ед./ час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	СРС, час.	Форма промежуточного контроля (экз./зачет)
1	1 ЗЕ / 36 час.	—	—	—	36	переаттестация (зачёт)
1	3 ЗЕ / 108 час.	2	—	6	100	зачёт
Итого	4 ЗЕ / 144 час.	2	—	6	136	переаттестация (зачёт), зачёт

Владимир 2015

## **1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

Важным фактором подготовки специалистов в области математического обеспечения и администрирования информационных систем является умение программировать, используя современные языки, знакомство с основными методами и современными технологиями программирования.

Цель курса: на базе языка Си (точнее, подмножества языка C++, не включающего объектно-ориентированное программирование) – одного из ключевых для эволюции языков программирования – ознакомление с основными методологиями и парадигмами программирования, семантикой и синтаксисом императивного языка программирования и средой разработки приложений.

Цель обусловила и задачи курса:

- понимание отличий различных парадигм программирования;
- изучение стандартов общего представления алгоритмов на языках программирования;
- изучение семантики и синтаксиса языка программирования;
- знакомство с особенностями сред разработки программного обеспечения;
- приобретение навыков разработки алгоритмов, для решения поставленной задачи, и навыков работы в команде.

## **2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО**

Дисциплина «Основы программирования» относится к базовой части ОПОП подготовки бакалавров по направлению «Математическое обеспечение и администрирование информационных систем».

Изучение дисциплины предполагает наличие у студентов навыков построения алгоритмов, знаний понятий алгоритмизации, которые могут быть получены в рамках дисциплин «Алгоритмы и анализ сложности», изучаемой параллельно в том же семестре, а также фундаментальных знаний в области информатики.

По дисциплине предусмотрена частичная переаттестация в объёме 1 зачётной единицы, так как часть содержания курса студенты осваивают в ходе обучения на профильных специальностях среднего профессионального образования.

Дисциплина формирует знания и навыки, необходимые в ходе дальнейшего обучения и в практической деятельности квалифицированного специалиста. В рамках учебного процесса может быть использована в ходе изучения дисциплин:

- Вычислительная математика
- Объектно-ориентированное программирование
- Структуры и алгоритмы компьютерной обработки данных
- Базы данных
- Программирование для Интернет и веб-дизайн
- Функциональное программирование
- Распределённая обработка информации / Параллельное программирование

Кроме того, полученные навыки будут необходимы при изучении дисциплин, в рамках которых требуется описывать модели на языке программирования и выполнять автоматизированные эксперименты, при прохождении практик и подготовке выпускной квалификационной работы.

## **3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

В результате изучения дисциплины студент должен частично овладеть следующими компетенциями:

- ОК-7, способностью к самоорганизации и самообразованию;
- ОПК-3, готовностью анализировать проблемы и направления развития технологий программирования;
- ОПК-7, способностью использовать знания основных концептуальных положений функционального, логического, объектно-ориентированного и визуального направлений программирования, методов, способов и средств разработки программ в рамках этих направлений;
- ОПК-11, готовностью использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях.

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

1) Знать:

- семантику и синтаксис изученных элементов языка программирования С/С++ (ОПК-7);
- парадигмы и методологии программирования на процедурных языках (ОПК-3);
- проблемы и направления развития технологий программирования (ОПК-3).

2) Уметь:

- самостоятельно разработать алгоритм и запрограммировать его, используя возможности процедурного языка программирования (ОК-7, ОПК-7, ОПК-11);
- выполнять отладку программ, используя возможности среды разработки (ОПК-7, ОПК-11).

3) Владеть:

- навыками самостоятельного изучения языков программирования (ОК-7);
- навыками работы в средах программирования (ОПК-7);
- навыками написания программ, реализующих алгоритмы на изученных языках программирования (ОПК-7).

#### 4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 4 зачетных единицы, 144 часа.

№ п/п	Раздел (тема) дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)						Объем учебной работы, с применением интерактивных методов (в часах / %)	Формы текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные работы	Контрольные работы	СРС	КП / КР		
1	Раздел 2. Базовые сведения о языке С++	1	1-6	-	-	-	-	12	-	-	
2	Раздел 3. Пользовательские типы данных языка С++	1	1-6	-	-	-	-	12	-	-	
3	Раздел 4. Работа с функциями и файлами. Дополнительные возможности	1	1-6	-	-	-	-	12	-	-	
<b>ИТОГО I СЕМЕСТР</b>				-	-	-	-	36	-	-	переаттестация (зачёт)

№ п/п	Раздел (тема) дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)						Объем учебной работы, с применением интерактивных методов (в часах / %)	Формы текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные работы	Контрольные работы	СРС	КП / КР		
4	Раздел 1. Введение в программирование. Среда разработки программного обеспечения	1	1-18	-	-	1	-	12	-	1 / 100%	Рейтинг-контроль №1
5	Раздел 2. Базовые сведения о языке C++	1	1-18	1	-	1	-	22	-	1 / 75%	
6	Раздел 3. Пользовательские типы данных языка C++	1	1-18	-	-	2	-	33	-	1 / 40%	
7	Раздел 4. Работа с функциями и файлами. Дополнительные возможности	1	1-18	1	-	2	-	33	-	1 / 33%	Рейтинг-контроль №3
<b>ИТОГО I СЕМЕСТР</b>				<b>2</b>	<b>-</b>	<b>6</b>	<b>-</b>	<b>100</b>	<b>-</b>	<b>4 / 50%</b>	<b>зачёт</b>
<b>Всего</b>				<b>2</b>	<b>-</b>	<b>6</b>	<b>-</b>	<b>136</b>	<b>-</b>	<b>4 / 50%</b>	<b>перееаттестация (зачёт), зачёт</b>

### Темы лекций.

#### **Раздел 2. Базовые сведения о языке C++.**

1. Структура программы. Базовые типы данных. Операции. Ввод/вывод данных.
2. Операторы языка C/C++.

#### **Раздел 3. Пользовательские типы данных языка C++.**

1. Массивы. Работа со строками.
2. Указатели и ссылки. Работа с динамической памятью.
3. Структуры. Составные типы данных.

#### **Раздел 4. Работа с функциями и файлами. Дополнительные возможности.**

1. Работа с функциями.
2. Работа с файлами. Функции работы с потоками.
3. Пространства имён. Обработка исключительных ситуаций. Проблемы и направления развития технологий программирования.

### Лабораторный практикум

- 1) Среда разработки Microsoft Visual Studio (1 ч).
- 2) Ввод/вывод данных. Операторы C++ (1 ч).
- 3) Работа с массивами. Программирование задач с использованием структур. (2 ч).
- 4) Функции (2 ч).

## 5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В рамках лекционного курса и лабораторных занятий:

- Технология проблемного обучения (case study). При рассмотрении вопросов практического применения рассмотренного теоретического материала, используется диалог со студентами на предмет возможных способов решения поставленной задачи.
- Обучение в малых группах (выполнение лабораторных работ в группах из двух или трёх человек);
- Технология развития критического мышления (прививание студентам навыков критической оценки разработанных ими алгоритмов);
- Мультимедиа-технологии (проведение лекционных и лабораторных занятий с использованием проекторов и других мультимедийных устройств).

В рамках самостоятельной работы:

- Технология проблемного обучения (case study).

## 6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

### Самостоятельная работа студентов при подготовке к переекзамену

Направлена на восстановление и упорядочение знаний студентов в области программирования, полученных на уровне среднего профессионального образования. В ходе самостоятельной работы студенты должны рассмотреть следующие вопросы:

1. Классификация парадигм и языков программирования.
2. Методология разработки программного обеспечения.
3. Среды разработки.
4. Тестирование и отладка программного кода.
5. Типы данных, переменные.
6. Операторы (в том числе условный, цикла с предусловием, цикла с постусловием).
7. Операции.
8. Подпрограммы, функции, процедуры.
9. Массивы.
10. Области видимости переменных.

### Рейтинг-контроль №1

Призван вывить знания студентами базовых сведений о языке C/C++.

Список вопросов (открытый):

*Вопрос 1. Могут ли совпадать имена объектов языка с ключевыми словами?*

*Вопрос 2. Какие из приведенных ниже слов являются ключевыми словами языка C++ ?*

- *signed*
- *sizeof*
- *then*
- *while*
- *repeat*
- *do*
- *until*
- *label*
- *int*

*Вопрос 3. Что означает наличие в программе конструкции #define BOOL int?*

*Вопрос 4. Исходная программа представляет собой набор элементов (дополнить список):*

- *директивы препроцессора*
- *объявления*

- указания компилятору

Вопрос 5. Дать определение понятию «объявление переменной».

Вопрос 6. С выполнения какой функции начинается выполнение программы на C++?

- *main*
- *first*
- *go*
- *другое*

Вопрос 7. Базовые типы языка C++:

- *char*
- *void*
- *double*
- *short int*
- *int*
- *float*
- *long int*
- *long double*

Вопрос 8. Что означает спецификатор типа *unsigned*?

Вопрос 9. Вывести на экран значение целочисленной переменной *int x*:

Вопрос 10. Как работает префиксная операция ++? Какое значение будет записано в память и какое использоваться в выражении? Привести пример с пояснениями.

Вопрос 11. Пояснить принцип работы оператора *break*.

Вопрос 12. Синтаксис оператора цикла с предусловием:

Вопрос 13. Проставить порядок действий и вычислить значение *k*:

Вопрос 14. Проставить порядок действий и записать значения всех переменных выражения:

Вопрос 15. Что будет напечатано после выполнения кода:

```
int a=10, b=3, c=-2, k;
k = a<b ? printf ("a<b") : (a<c)?printf ("a<c") :printf ("a>=c") ;
printf("\n k=%d", k);
```

Вопрос 16. Посчитать сумму элементов последовательности  $S = \sum_{i=1}^{100} \frac{1}{2i^2}$ . При написании программы комментарии обязательны.

## Рейтинг-контроль №2

Призван вывить знания студентами принципов работы с пользовательскими типами в языке C/C++.

Список вопросов (открытый):

Вопрос 1. Дать определение массива с точки зрения хранения.

Вопрос 2. Привести пример обращения к элементу массива

Вопрос 3. Дано объявление: *int (\*a)[5]*; Какой тип данных имеет переменная *a*?

Вопрос 4. Верно ли утверждение, что *a[i] = i[a]*, если *a* - массив?

Вопрос 5. Каково название функции копирования строк?

Вопрос 6. Способы инициализации указателей.

Вопрос 7. С помощью какой функции происходит освобождение памяти, выделенной с помощью оператора *new*?

Вопрос 8. *int \*m = new int(10)*; Пояснить работу приведенного кода.

Вопрос 9. Ключевое слово для определения типа структура?

Вопрос 10. Что такое вложенные структуры? Привести пример.

Вопрос 11. Объявить структуру, содержащую информацию о книге: название, автор, год издания, количество страниц.

Вопрос 13. Объявите массив из пяти переменных, типа *int*, и заполните его (программно) первыми пятью целыми нечетными числами.

Вопрос 14. Объявить ссылку на переменную *int a = 5*.

Вопрос 15. С помощью какого ключевого слова можно объявить перечисление?

*Вопрос 16. Посчитать количество положительных элементов в массиве до первого отрицательного. Программу писать с используемыми заголовочными файлами, объявлением главной функции и комментариями.*

### **Рейтинг-контроль №3**

Призван вывить знания студентами принципов работы с функциями и файлами в языке C/C++.

Список вопросов (открытый):

*Вопрос 1. Параметры функций являются...*

- Локальными
- Глобальными
- Статическими

*Вопрос 2. Написать функцию сложения двух целых чисел, параметры в функцию передать по указателю. Сделать вызов этой функции.*

*Вопрос 3. inline функции. Условия, когда функция может быть inline.*

*Вопрос 4. Ключевое слово для объявления шаблона функции.*

*Вопрос 5. Написать функцию копирования подстроки строки S1, начиная с символа номер k1 до символа k2, с организацией правильности задания k1. Если k2 больше длины строки, то копировать до конца строки. Строка символов вводится с клавиатуры. Программу писать с используемыми заголовочными файлами, объявлением главной функции и комментариями.*

### **Переаттестация (зачёт)**

Призвана вывить полученные студентами на предыдущем уровне образования знания основ программирования. Проводится в форме решения задачи на любом из известных студенту языков. Примеры задач:

1. *Описать и заполнить одномерный массив целых чисел. Найти сумму неотрицательных элементов массива. Отобразить результат.*
2. *Описать и заполнить одномерный массив целых чисел. Найти сумму положительных элементов массива. Отобразить результат.*
3. *Описать и заполнить одномерный массив целых чисел. Найти произведение отрицательных элементов массива. Отобразить результат.*
4. *Описать и заполнить одномерный массив целых чисел. Найти произведение положительных элементов массива. Отобразить результат.*
5. *Описать и заполнить одномерный массив целых чисел. Найти максимальный из положительных элементов массива (значение и номер). Отобразить результат.*
6. *Описать и заполнить одномерный массив целых чисел. Найти минимальный из положительных элементов массива (значение и номер). Отобразить результат.*
7. *Описать и заполнить одномерный массив целых чисел. Найти сумму чётных (по номеру элемента) элементов массива. Отобразить результат.*
8. *Описать и заполнить одномерный массив целых чисел. Найти сумму чётных (по значению элемента) элементов массива. Отобразить результат.*
9. *Описать и заполнить одномерный массив целых чисел. Найти максимальный из нечётных (по номеру элемента) элементов массива. Отобразить результат.*
10. *Описать и заполнить одномерный массив целых чисел. Найти минимальный из нечётных (по значению элемента) элементов массива. Отобразить результат.*
11. *Описать и заполнить двумерный массив целых чисел. Найти максимальный из чётных (по значению элемента) элементов массива. Отобразить результат.*
12. *Описать и заполнить двумерный массив целых чисел. Найти максимальные элементы в каждой из строк массива (номера и значения). Отобразить результат.*
13. *Описать и заполнить двумерный массив целых чисел. Найти минимальные элементы в каждом из столбцов массива (номера и значения). Отобразить результат.*

14. Описать и заполнить двумерный массив целых чисел. Найти сумму элементов в каждой из строк массива. Отобразить результат.
15. Описать и заполнить двумерный массив целых чисел. Найти произведение ненулевых элементов в каждом из столбцов массива. Отобразить результат.
16. Описать и заполнить двумерный массив целых чисел. Заменить у всех ненулевых элементов массива знак на противоположный. Отобразить результат.
17. Описать и заполнить двумерный массив целых чисел. Найти сумму элементов главной диагонали (массив не обязательно должен быть квадратным). Отобразить результат.
18. Описать и заполнить двумерный массив целых чисел. Найти сумму элементов побочной диагонали (массив не обязательно должен быть квадратным). Отобразить результат.
19. Описать и заполнить двумерный массив целых чисел. Найти сумму максимальных элементов каждой из строк массива. Отобразить результат.
20. Описать и заполнить двумерный массив целых чисел. Найти сумму минимальных элементов каждого из столбцов массива. Отобразить результат.

### 1 семестр

#### Самостоятельная работа студентов

Вопросы для контроля самостоятельной работы (используются во время защиты лабораторных работ, для самоконтроля и для формулирования домашних заданий):

- 1) Можно ли использовать ключевые слова для именования объектов программы?
- 2) Перечислите основные элементы программы.
- 3) В чем отличия определения и объявления объектов программы?
- 4) Перечислите базовые типы данных языка и их характеристики.
- 5) Поясните принцип работы префиксной и постфиксной операций увеличения.
- 6) Каким образом можно, не используя операцию умножения, умножить число на 4.
- 7) Укажите порядок вычисления следующих выражений, задав полную скобочную структуру:

$a = b + c * d << 2 \& 8$

$a = -1 ++ b -- -5$

$a == b || a == c \&\& c < 5$

$c = x != 0$

$a \& 077 != 3$

$a = b == c ++$

$a = b = c = 0$

$a-b, c=d$

- 8) Перечислите циклические операторы: синтаксис и семантику.
- 9) Назначение блочного оператора.
- 10) Поясните принцип работы операторов *break* и *continue*.
- 11) В операторе-переключателе является ли обязательной метка *default*?
- 12) Следующий цикл *for* перепишите с помощью оператора *while*:

```
for (i=0; i<max_length; i++)
    if (input_line[i] == '?') quest_count++;
```

Запишите цикл, используя в качестве его управляющей переменной указатель так, чтобы условие имело вид  $*p == '?'$ .

- 13) Дайте определение массива с точки зрения хранения.
- 14) Приведите общий принцип обращения к элементам массива.
- 15) С какого числа начинается нумерация элементов массива? Можно ли нумерацию изменить?
- 16) Каким образом определить размер массива?
- 17) Объявить трехмерный массив и изобразить его с точки зрения языка C++.
- 18) Объявить двумерный массив вещественных элементов размером 5 строк и 4 столбца.

Далее:

- Проинициализировать при объявлении;
  - Организовать заполнение массива с клавиатуры;
  - Организовать заполнение случайными числами.
- 19) Посчитать сумму элементов массива.
  - 20) Может ли элементом массива быть массив?
  - 21) Что такое строка с точки зрения языка C++?



- 22) В чем отличие строки от массива символов?
- 23) Можно ли со строкой работать поэлементно?
- 24) Написать программу, вычисляющую сумму цифр в строке вида "1ab3c405". Ввод строки организовать с клавиатуры.
- 25) Написать программу, удаляющую все цифры из символьной строки.
- 26) Написать фрагмент кода, осуществляющий смену двух строк `str1` и `str2`, если они одинаковой длины.
- 27) Что такое указатель?
- 28) Перечислите проблемы, которые могут возникнуть при работе с инициализированными указателями.
- 29) Назначение нетипизированного указателя? Каким образом можно объявить нетипизированный указатель?
- 30) Будет ли корректно работать следующий код:
- ```
int a = 5;
int *pf = &a;
float *p;
p = pf;
```
- Если код работает некорректно, то внести исправления.
- 31) Объявить массив из трех указателей на вещественные переменные. Задать значения переменных через указатели.
- 32) Разместить в динамической памяти одномерный массив, двумерный массив.
- 33) Поясните, что объявлено, проинициализировать все объявленные переменные и нарисовать картинки в памяти и с точки зрения языка Си.
- ```
int (*pM) [3];
int *(*pMM) [2];
int m[2][3];
```
- 34) Написать фрагмент программы, используя оператор выделения динамической памяти `new`. В программе должен выполняться захват памяти для пяти символов, ввод строки с клавиатуры и освобождение захваченной памяти.
- 35) Что такое ссылка?
- 36) Объявить ссылку на константу.
- 37) Назовите отличия структуры от массива?
- 38) Назовите операции доступа к полям структуры по указателю и через объект.
- 39) Назовите назначение битовых полей и особенности работы с ними.
- 40) Что такое функция?
- 41) Для чего применяются прототипы функций?
- 42) В чем отличия объявления и определения функции?
- 43) Что означает передача аргумента в функцию по ссылке? Приведите пример.
- 44) Написать функцию ввода с клавиатуры переменной структурного типа (структура Студент содержит поля: ФИО, дата рождения, успеваемость по 5 дисциплинам). Сформированные структуры из функции получать следующими способами:
- В качестве параметра функции.
  - В качестве возвращаемого значения.
- Продемонстрировать работу функции.
- 45) Вернуть из функции массив указателей на целочисленные переменные. Продемонстрировать работу функции.
- 46) Написать шаблонную функцию для нахождения суммы элементов числового массива, массив передавать в функцию в качестве параметра. Привести пример использования функции для нескольких числовых типов.
- 47) Объявить указатель на функцию, имеющую следующий прототип:
- ```
int func(char *, int);
```
- 48) Каким образом можно передавать одномерные массивы в функции? Приведите примеры.
- 49) Поясните принцип передачи многомерных массивов в функции?
- 50) Что такое операторные функции? Назначение перегрузки операторов?
- 51) Дайте определение перегруженной функции.
- 52) За счет какого механизма возможна перегрузка функций в C++?

- 53) Объявить *inline* функцию, например, вычисления квадрата числа. Объявить указатель на эту функцию. Вызвать *inline* функцию через указатель. Будет ли функция в этом случае трактоваться как встраиваемая?
- 54) Что такое файл?
- 55) Приведите отличия логического и физического файлов.
- 56) Перечислите типы файлов?
- 57) Приведите классификацию файлов по способам доступа к информации.
- 58) Какие действия необходимо сделать для работы с файлом?
- 59) Какая функция используется для открытия файла? Опишите параметры функции.
- 60) Каким образом можно определить, достигнут ли конец файла?
- 61) Написать программу, которая считывает из текстового файла три предложения и выводит их в обратном порядке.
- 62) Если требуется осуществить быстрое копирование файлов неизвестной структуры, какого типа файл нужно использовать?
- 63) Написать программу, которая считывает текст из файла и выводит на экран, после каждого предложения добавляя, сколько раз встречалось в нем введенное с клавиатуры слово.

### Зачёт

Предполагает письменное решение студентом практической задачи на языке C/C++. Конкретная задача выбирается случайным образом из базы задач, имеющейся у преподавателя. Во время решения задачи студент может пользоваться справочной литературой и конспектами. Уровень решения определяет выставляемую оценку. При этом студентом предварительно должны быть выполнены все задания лабораторного практикума.

### Примеры задач:

1. Описать и заполнить двумерный массив структур. Структура должна содержать строковое поле и поле целого типа. Найти элементы в каждом из столбцов массива (номера и значения) с минимальным значением целочисленного поля. Отобразить результат. При работе с массивом обращаться к его элементам через указатели.
2. Описать и заполнить двумерный массив структур. Структура должна содержать строковое поле и поле целого типа. Найти сумму целочисленных полей элементов в каждой из строк массива. Отобразить результат. При работе с массивом обращаться к его элементам через указатели.
3. Описать и заполнить двумерный массив структур. Структура должна содержать строковое поле и поле целого типа. Найти произведение ненулевых целочисленных полей элементов в каждом из столбцов массива. Отобразить результат. При работе с массивом обращаться к его элементам через указатели.
4. Описать и заполнить двумерный массив структур. Структура должна содержать строковое поле и поле целого типа. Заменить у всех ненулевых целочисленных полей элементов массива знак на противоположный. Отобразить результат. При работе с массивом обращаться к его элементам через указатели.
5. Описать и заполнить двумерный массив структур. Структура должна содержать строковое поле и поле целого типа. Найти сумму целочисленных полей элементов главной диагонали (массив не обязательно должен быть квадратным). Отобразить результат. При работе с массивом обращаться к его элементам через указатели.
6. Описать и заполнить двумерный массив структур. Структура должна содержать строковое поле и поле целого типа. Найти сумму целочисленных полей элементов побочной диагонали (массив не обязательно должен быть квадратным). Отобразить результат. При работе с массивом обращаться к его элементам через указатели.
7. Описать и заполнить двумерный массив структур. Структура должна содержать строковое поле и поле целого типа. Найти сумму максимальных целочисленных полей элементов каждой из строк массива. Отобразить результат. При работе с массивом обращаться к его элементам через указатели.
8. Описать и заполнить двумерный массив структур. Структура должна содержать строковое поле и поле целого типа. Найти сумму минимальных целочисленных полей элементов каждого из столбцов массива. Отобразить результат. При работе с массивом обращаться к его элементам через указатели.

## 7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) основная литература:

1. Основы алгоритмизации и программирования: Учебное пособие / В.Д. Колдаев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ; ИНФРА-М, 2015. - 416 с.: ил.; 60x90 1/16. - (Профессиональное образование). (переплет) ISBN 978-5-8199-0279-0
2. Программирование на языке Си/А.В.Кузин, Е.В.Чумакова - М.: Форум, НИЦ ИНФРА-М, 2015. - 144 с.: 70x100 1/16. - (Высшее образование) (Обложка) ISBN 978-5-00091-066-5
3. Царев, Р. Ю. Программирование на языке Си [Электронный ресурс] : учеб. пособие / Р. Ю. Царев. – Красноярск : Сиб. федер. ун-т, 2014. – 108 с. - ISBN 978-5-7638-3006-4

б) дополнительная литература:

1. Программирование на языке высокого уровня. Программирование на языке Object Pascal: Учебное пособие / Т.И. Немцова; Под ред. Л.Г. Гагариной. - М.: ФОРУМ; ИНФРА-М, 2015. - 496 с.: ил.; 60x90 1/16. - (ПО). ISBN 978-5-8199-0372-8
2. Введение в среду визуального программирования Turbo Delphi. Ч.3 [Электронный ресурс] : Учеб. пособие / С. В. Борисов, С. С. Комалов, О. Б. Пашенко, И. Л. Серебрякова. - М. : Издательство МГТУ им. Н. Э. Баумана, 2013
3. Задачи по программированию [Электронный ресурс] / С. М. Окулов [и др.] ; под ред. С. М. Окулова. - 2-е изд., испр. (эл.).-М. : БИНОМ. Лаборатория знаний, 2014. - 823 с. : ил. - ISBN 978-5-9963-2372-2

в) периодические издания:

1. Computerworld Россия, ISSN: 1560-5213.
2. Мир ПК, ISSN: 0235-3520.
3. CHIP Россия, ISSN: 1609-4212.

г) интернет-ресурсы:

1. MSDN. Возвращение к С++ (современный С++). // Режим доступа: <https://msdn.microsoft.com/ru-ru/library/hh279654.aspx>
2. MSDN. Справочник по языку С++. // Режим доступа: <https://msdn.microsoft.com/ru-ru/library/3bstk3k5.aspx>
3. MSDN. Visual Studio. // Режим доступа: <https://www.visualstudio.com/>


## 8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

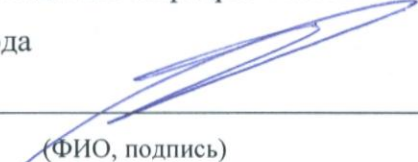
Компьютерные классы для проведения лабораторных занятий. Компьютеры должны иметь выход в Интернет для получения доступа к справочной информации Microsoft Developer Network.

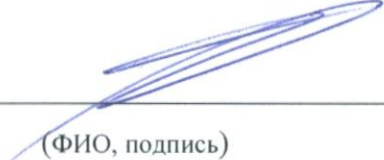
Среда разработки Microsoft Visual Studio.

Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО по направлению 02.03.03 Математическое обеспечение и администрирование информационных систем

Рабочую программу составил доцент каф. ФиПМ Лексин А.Ю.   
(ФИО, подпись)

Рецензент  
(представитель работодателя) ген. директор ООО "РС Сервис" Ю.В. Квицов   
(место работы, должность, ФИО, подпись)

Программа рассмотрена и одобрена на заседании кафедры ФиПМ  
Протокол № 11А от 17.04.15 года  
Заведующий кафедрой \_\_\_\_\_ Аракелян С.М.  
  
(ФИО, подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии направления 02.03.03 Математическое обеспечение и администрирование информационных систем  
Протокол № 11А от 17.04.15 года  
Председатель комиссии \_\_\_\_\_ Аракелян С.М.  
  
(ФИО, подпись)

### ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Рабочая программа одобрена на \_\_\_\_\_ учебный год  
Протокол заседания кафедры № \_\_\_\_\_ от \_\_\_\_\_ года  
Заведующий кафедрой \_\_\_\_\_

Рабочая программа одобрена на \_\_\_\_\_ учебный год  
Протокол заседания кафедры № \_\_\_\_\_ от \_\_\_\_\_ года  
Заведующий кафедрой \_\_\_\_\_

Рабочая программа одобрена на \_\_\_\_\_ учебный год  
Протокол заседания кафедры № \_\_\_\_\_ от \_\_\_\_\_ года  
Заведующий кафедрой \_\_\_\_\_