

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



УТВЕРЖДАЮ
 Проректор
 по образовательной деятельности
 А.А.Панфилов
 « 08 » 2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ОСНОВЫ ПРОГРАММИРОВАНИЯ
 (наименование дисциплины)

Направление подготовки 02.03.02 Фундаментальная информатика и информационные технологии

Профиль/программа подготовки Мобильные и Интернет-технологии

Уровень высшего образования бакалавриат

Форма обучения очная, ускоренная

Семестр	Трудоемкость зач. ед./ час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	СРС, час.	Форма промежуточной аттестации (экзамен/зачет/зачет с оценкой)
1	5/180				180	Зачет (переаттестация)
1	7 / 252	36	36	36	108	Экзамен, 36
Итого	12 / 432	36	36	36	288	Зачет (переаттестация), Экзамен, 36

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель освоения дисциплины – приобретение навыков автоматизации профессиональных задач за счет разработки алгоритмов и компьютерных программ для их решения.

Задачи:

- понимание отличий различных парадигм программирования;
- изучение принципов разработки алгоритмов;
- 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;
- навыков работы в команде.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина «Основы программирования» относится к обязательной части блока I дисциплин (модулей) учебного плана.

Пререквизиты дисциплины: дисциплина опирается на знания предметов основной образовательной программы среднего (полного) общего образования: информатика и ИКТ.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП

Код формируемых компетенций	Уровень освоения компетенции	Планируемые результаты обучения по дисциплине характеризующие этапы формирования компетенций (показатели освоения компетенции)
1	2	3
ОПК-2	частичное	Знать: основные положения и концепции в области программирования; архитектура языков программирования; основная терминология в области программного обеспечения; Уметь: умеет осуществлять обоснованный выбор компьютерных/суперкомпьютерных методов и необходимого программного обеспечения при решении задач профессиональной деятельности; Владеть: навыками применения данных методов и программного обеспечения при решении конкретных задач.
ОПК-3	частичное	Знать: методы теории алгоритмов; методы системного и прикладного программирования; Уметь: умеет соотносить знания в области программирования; умеет осуществлять обоснованный выбор адекватных поставленной задаче базовых математических моделей; Владеть: навыками разработки программного обеспечения;
ОПК-4	частичное	Знать: стандарты, нормы и правила документирования программных продуктов и комплексов; Уметь: умеет соотносить знания в области программирования; умеет осуществлять обоснованный выбор адекватных поставленной задаче базовых математических моделей; Владеть: навыками документирования процесса создания информационных систем на разных стадиях жизненного цикла;
ПК-1	частичное	Знать: возможности современных и перспективных средств разработки программных продуктов, технических средств; типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения; Уметь: выбирать средства и варианты реализации программного обеспечения; Владеть: навыками согласования требований к программному обеспечению с заинтересованными сторонами; навыки оценки и согласования сроков выполнения поставленных задач; навыки разработки и согласования технических спецификаций на программное обеспечение; навыки формирования и предоставления отчетности в соответствии с установленными регламентами;
ПК-2	частичное	Знать: методы анализа, проектирования и разработки программного обеспечения; Уметь: использовать современные CASE-средства; Владеть: навыками проверки осуществимости функционирования и сопровождения программного средства;

4. ОБЪЕМ И СТРУКТУРА ДИСЦИПЛИНЫ

Трудоемкость дисциплины составляет 12 зачетных единиц, 432 часа.

№ п/п	Наименование тем и/или разделов/тем 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, форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные работы	СРС		
1.	Введение в дисциплину	1	1						
1.1.	Развитие языков программирования. Парадигмы программирования		2	2	–	4	0/0%		
	Развитие языков программирования. Парадигмы программирования		–	–	–	8		Переаттестация	
1.2.	Среда разработки программного обеспечения		–	–	2	6	2/100%		
	Среда разработки программного обеспечения		–	–	–	8		Переаттестация	
2.	Основы алгоритмизации		2-10						
2.1.	Понятие и основы свойства алгоритма. Способы записи алгоритмов.		2	2	2	8	2/33,3%		
	Понятие и основы свойства алгоритма. Способы записи алгоритмов.		–	–	–	10		Переаттестация	
2.2.	Типовые алгоритмические структуры		3	2	2	10	2/25%	Рейтинг-контроль №1	
	Типовые алгоритмические структуры		–	–	–	20		Переаттестация	
2.3.	Комбинированные алгоритмы		4-8	4	10	4	12	4/33,3%	
	Комбинированные алгоритмы		–	–	–	16		Переаттестация	
3.	Программирование на языке программирования высокого уровня		6-18						
3.1.	Введение в C++		6-7	4	–	–	6	2/33,3%	
	Введение в C++		–	–	–	10		Переаттестация	
3.2.	Операторы языка C++		8	2	–	2	4	2/50%	
	Операторы языка C++		–	–	–	16		Переаттестация	
3.3.	Массивы	8-18	2	20	4	12	4/33,3%		
	Массивы	–	–	–	21		Переаттестация		
3.4.	Строки	9-10	2	–	4	6	2/33,3%		
	Строки	–	–	–	8		Переаттестация		
3.5.	Указатели и ссылки	10-12	2	–	4	6	2/33,3%	Рейтинг-контроль №2	
3.6.	Составные типы данных: структуры, перечисления и объединения	11-14	2	–	4	6	2/33,3%		
3.7.	Работа с функциями	13-16	4	–	4	11	2/25%		
	Работа с функциями	–	–	–	30		Переаттестация		
3.8.	Файлы	15-18	4	–	4	10	2/25%		
	Файлы	–	–	–	24		Переаттестация		
4.	Структуры компьютерной обработки данных	17-18							
4.1.	Связные списки	17	2	–	–	8	2/100%		

№ п/п	Наименование тем и/или разделов/тем 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, форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные работы	СРС		
4.2.	Табличное хранение данных		18	2	-	-	8	2/100%	Рейтинг-контроль
Всего за 1 семестр:		1	18	36	36	36	288	32 / 35,5%	№3
Наличие в дисциплине КП/КР		-	-	-	-	-	-	-	экзамен
Итого по дисциплине		1	18	36	36	36	288	32 / 35,5%	экзамен, 36

Содержание лекционных занятий по дисциплине

Раздел 1. Введение в дисциплину

Тема 1 Развитие языков программирования. Парадигмы программирования

История развития языков программирования. Классификация языков программирования. Интерпретация и компиляция программ.

Раздел 2 Основы алгоритмизации

Тема 1 Понятие и основы свойства алгоритма

Алгоритм. Свойства алгоритма. Исполнитель алгоритма. Этапы подготовки вычислительных задач для их автоматического решения. Формы записи алгоритмов. Основной набор элементарных предписаний алгоритма (изучение ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения).

Тема 2 Типовые алгоритмические структуры

Основные типы простейших алгоритмических структур. Анализ сложности алгоритмов по времени исполнения и памяти.

Тема 3 Комбинированные алгоритмы

Вложенные циклы. Алгоритмы накопления сумм и произведений. Разветвления в теле цикла в том числе и с досрочным выходом из цикла. Итерационные циклы.

Раздел 3 Программирование на языке программирования высокого уровня

Тема 1 Введение в C++

Структура программы. Базовые типы данных. Операции и функции на базовых типах данных. Приоритеты операций. Основы ввода/вывода: Ввод/вывод функциями из библиотеки `stdio.h` и потоковый ввод/вывод из библиотеки `iostream`. Переменные: время жизни и область действия.

Тема 2 Операторы языка C++

Пустой оператор. Составной оператор. Оператор возврата `return`. Оператор-выражение. Условный оператор. Оператор-переключатель `switch`. Циклические операторы. Операторы `break`, `continue`.

Тема 3 Массивы

Объявление и определение массива. Доступ к элементам массива. Одномерные и многомерные массивы. Сечения массивов и их применение в программах. Массив с точки зрения хранения и с точки зрения C++. Массивы в статической и динамической памяти. Примеры решения задач с массивами.

Тема 4 Строки

Объявление строки. Строка как массив символов специальной структуры. Функции библиотеки `string.h`. Тип данных `string`. Работа со строками средствами библиотеки `string`.

Тема 5 Указатели и ссылки

Понятие указателя. Инициализация указателей. Операции с указателями. Ссылки. Константные указатели и константные ссылки. Указатели на константы и ссылки на константы и их применение в программах. Работа с динамической памятью.

Тема 6 Составные типы данных: структуры, перечисления и объединения

Работа со структурами: объявление типа данных, переменных, доступ к полям структуры через переменную и указатель на структуру. Перечисление. Объединение.

Тема 7 Работа с функциями

Прототип функции. Объявление и определение функции. Вызов функции. Способы возврата значений из функции. Способы передачи параметров в функции. Вызов функции. Встраиваемые функции. Параметры функции по умолчанию. Перегрузка функций. Перегрузка операторов. Шаблоны функции. Параметры, передаваемые через командную строку в main. Рекурсивные функции.

Тема 8 Файлы

Физический и логический файл. Виды файлов и варианты доступа в зависимости от вида. Работа с файлами через указатель на FILE. Операции с файлами: открытие, закрытие, чтение и запись в файл. Работа с файлами через потоки. Обработка исключительных ситуаций при работе с файлами.

Раздел 4 Структуры компьютерной обработки данных

Тема 1 Связные списки

Односвязные списки. Двусвязные списки. Стек. Очередь. Бинарное дерево.

Тема 2 Табличное хранение данных

Таблицы с прямой адресацией. Хеш-таблицы.

Содержание практических занятий по дисциплине

Раздел 2 Основы алгоритмизации

Тема 1 Понятие и основы свойства алгоритма. Способы записи алгоритмов.

1. Изучение записи алгоритмов в виде блок-схем. Процесс построения алгоритма решения задачи на примере линейного алгоритма.

Тема 2 Типовые алгоритмические структуры

2. Разветвленные и циклические алгоритмы.

Тема 3 Комбинированные алгоритмы

3. Вложенные циклы.

4. Алгоритмы накопления сумм и произведений.

5. Итерационные циклы. Приближенное вычисление значений некоторых функций.

Раздел 3 Программирование на языке программирования высокого уровня

Тема 3 Массивы

6. Разветвления в теле цикла.

7. Сортировка числовых массивов.

8. Комбинированные алгоритмы работы с одномерными и двумерными массивами.

9. Алгоритмы по работе с векторами и матрицами.

Содержание лабораторных занятий по дисциплине

Раздел 1. Введение в дисциплину

Тема 1 Среда разработки программного обеспечения

1. Разработка приложения в среде Microsoft Visual Studio.

Раздел 2 Основы алгоритмизации

Тема 1 Понятие и основы свойства алгоритма. Способы записи алгоритмов.

2. Реализация линейных алгоритмов. Разработка алгоритма на языке программирования C++

Тема 2 Типовые алгоритмические структуры

3. Реализация разветвленных алгоритмов.

4. Реализация циклических алгоритмов.

Тема 3 Комбинированные алгоритмы

5. Реализация комбинированных циклических алгоритмов: вложенные циклы, циклические алгоритмы с ветвлением внутри цикла.

Раздел 3 Программирование на языке программирования высокого уровня

Тема 2 Операторы языка C++

6. Реализация комбинированных циклических алгоритмов: алгоритмов накопления сумм и произведений. Оптимизация алгоритмов.

Тема 3 Массивы

7. Работа с одномерными и многомерными массивами.

Тема 4 Строки

8. Работа с символами и строками.

Тема 5 Указатели и ссылки

9. Работа с указателями и ссылками. Допустимые операции с указателями и ссылками. Работа с динамической памятью.

Тема 6 Составные типы данных: структуры, перечисления и объединения

10. Работа со структурным типом данных.

Тема 7 Работа с функциями

11. Функция. Способы передачи параметров в функцию. Реализация набора функций векторно-матричной арифметики. Реализация рекурсивной функции.

Тема 8 Файлы

12. Разработка программы по работе с файлом через указатель на FILE и через потоки.

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В преподавании дисциплины «Основы алгоритмизации и программирование» используются разнообразные образовательные технологии как традиционные, так и с применением активных и интерактивных методов обучения.

Активные и интерактивные методы обучения:

- Интерактивная лекция (все лекции);
- Групповая дискуссия (тема №4.1, 4.2);
- Тренинг (тема №1.2);
- Анализ ситуаций (тема №3.1, 3.3, 3.7);
- Разбор конкретных ситуаций (тема № 3.6);
- Проектная работа (тема №3.7).

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

Текущий контроль успеваемости проводится по всем видам занятий с использованием рейтинговой системы.

А. Вопросы для рейтинг-контроля.

Вопросы для рейтинг-контроля №1.

1. Дайте определение понятию алгоритм.

2. Выберите из списка правильный вариант утверждения: «Линейный алгоритм – это..

- А) вычислительная схема, которая содержит не одну, а несколько возможных ветвей решения;
- Б) вычислительная структура, при которой все предписания выполняются в строго линейной последовательности друг за другом;
- В) схема разветвлённой структуры, в которой одна ветвь операции сравнения является обратной связью.

3. Исправьте ошибку(и) в программе и объясните причину их возникновения.

```
#include <stdafx.h>
#include <conio.h>
#include <math.h>
#define <math.h>
int main()
{
int a;
double x;
int y;
y=pow(a,2)+(x/4);
printf("y=",y);
}
```

4. Выберите правильный вариант записи команды в программе.

- А. $Y = \sqrt{x^2 + \text{abs}(3-2) / \text{pow}((1-x), 2)}$
- Б. $y = \sqrt{(\text{pow}(x, 2) + (\text{abs}(3) - 2)) / \text{pow}(1-x, 2)}$
- В. $Y = \text{SQRT} * \text{POW}(x, 2) + \text{ABS}(3 - 2) / \text{POW}(1 - x, 2)$
- Г. Другой вариант

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

7. Вывести на экран значение целочисленной переменной int x.

Вопросы для рейтинг-контроля №2.

1. Дайте определение понятию массив.

2. Выберите из списка правильный вариант утверждения: «Указатели – это..

- А) переменные, предназначенные для хранения в них адресов каких-либо объектов программы;
- Б) тип данных, указывающий на определенную ячейку памяти;
- В) переменные, указывающие на местонахождение конкретного параметра в программе.

3. Составьте программу на языке C++, позволяющую отсортировать целочисленный одномерный массив, состоящий из 14 элементов по четным и нечетным позициям и вывести на экран элементы, имеющие нечетную позицию. (В ответе должна быть построена блок-схема).

4. Выберите правильный вариант записи команды в программе.

А. `int mas{12,5,8,9}`

Б. `int mas[12], int mas[4]`

В. int mas[8]

Г. Другой вариант

5. Составьте программу на языке C++, позволяющую находить сумму элементов матрицы (5*5), лежащих ниже главной диагонали. (В ответе должна быть построена блок-схема).

6. Дайте определение понятию «утечка памяти».

7. Дать определение массива с точки зрения хранения.

Вопросы для рейтинг-контроля №3.

1. Дайте определение понятию массив.

2. Выберите из списка правильный вариант утверждения: «Указатели – это..»

А) переменные, предназначенные для хранения в них адресов каких-либо объектов программы;

Б) тип данных, указывающий на определенную ячейку памяти;

В) переменные, указывающие на местонахождение конкретного параметра в программе.

3. Составьте программу на языке C++, позволяющую отсортировать целочисленный одномерный массив, состоящий из 14 элементов по четным и нечетным позициям и вывести на экран элементы, имеющие нечетную позицию. (В ответе должна быть построена блок-схема).

4. Выберите правильный вариант записи команды в программе.

А. int mas{ 12,5,8,9}

Б int mas[12], int mas[4]

В. int mas[8]

Г. Другой вариант

5. Составьте программу на языке C++, позволяющую находить сумму элементов матрицы (5*5), лежащих ниже главной диагонали. (В ответе должна быть построена блок-схема).

6. Дайте определение понятию «утечка памяти»

7.. Выберите из списка правильный вариант утверждения: «Циклический алгоритм – это...»

А) вычислительная схема, которая содержит не одну, а несколько возможных ветвей решения;

Б) вычислительная структура, при которой все предписания выполняются в строго линейной последовательности друг за другом;

В) схема разветвлённой структуры, в которой одна ветвь операции сравнения является обратной связью.

8. Составьте программу, которая создает два текстовых файла: в первом файле должна содержаться количество студентов в группе(22), а второй должен содержать отредактированную информацию (20)

Б. Вопросы к экзамену.

1. Понятие алгоритма. Алгоритмизация. Основные этапы разработки алгоритма.

2. Способы записи алгоритма. Блок-схема. Тестирование и отладка программы.

3. Элементы блок-схемы алгоритма: оператор «процесс» и его функции при разработке блок-схемы.

4. Элементы блок-схемы алгоритма: оператор «модификация» и его функции при составлении блок-схемы алгоритма.

5. Элементы блок-схемы алгоритма: оператор «условие» и его применение при составлении блок-схемы.

6. Алгоритмические структуры: линейный алгоритм, основные требования к записи данного вида алгоритма.

7. Алгоритмические структуры: разветвляющийся алгоритм, описание алгоритма, ветвление алгоритма.

8. Алгоритмические структуры: цикл с предусловием, описание алгоритма, форма записи, задание параметров цикла.

9. Алгоритмические структуры: цикл с постусловием, описание алгоритма, форма записи.

10. Алгоритмические структуры: вложенные циклы, правила определения порядка вложения циклов в алгоритме, понятие внешнего и внутреннего циклов.

11. Операторы условия в языке C++: конструкция if...else. Форма записи. Применение при разработке программы.

12. Операторы переключатели в языке C++: оператор switch...case. Форма записи. Применение при разработке программы.

13. Побитовые операции: сложение, вычитание, умножение и т.д.

14. Логические операции.

15. Оператор sizeof. Функции в программе.

16. Операторы ввода-вывода данных: printf(), scanf().

17. Поточковый ввод-вывод данных: cin>>, cout>>.

18. Структура программы на языке C++. Основные элементы структуры.

19. Библиотека <stdafx.h>. Параметры применения и функции.

20. Библиотека <stdio.h>. Форма записи и функции.
21. Библиотека <conio.h>. Форма записи и функции.
22. Библиотека <math.h>. Форма записи и функции.
23. Библиотека <iostream>. Форма записи и функции.
24. Массив: понятие, форма записи, инициализация массивов.
25. Одномерный массив: понятие, форма записи, способ записи.
26. Двумерный массив: понятие, форма записи, способ записи.
27. Сортировка массива. Методы сортировки: бинарный поиск, «пузырек».
28. Указатели: понятие, форма записи, вызов в программе и функции.
29. Статический массив. Форма записи.
30. Динамический массив. Форма записи.
31. Понятие «утечка памяти».
32. Понятие «индексация массива». Применение в программировании.
33. Операция упорядочивание массива по убыванию. Принцип действия в программе.
34. Операция упорядочивание массива по возрастанию.
35. Вспомогательный алгоритм: понятие, форма записи, структура.
36. Функция: понятие, форма записи, структура.
37. Фактические параметры функции.
38. Формальные параметры функции.
39. Фактические аргументы.
40. Прототипы функции: понятие, применение в программировании.
41. Способы передачи параметров в функцию. Ссылки.
42. Вызов функции в программе. Правила оформления функции в программе.
43. Описание функции.
44. Объявление функции.
45. Файлы: определение, классификация, сфера применения в программировании.
46. Форма записи файла в программе.
47. Текстовые файлы.
48. Двоичные файлы.
49. Основные функции для работы с файлами.
50. Подключение стандартных библиотек для ввода-вывода файлов.
51. Задача.

В. Вопросы к зачету (перееаттестации)

1. Этапы решения задач на компьютере.
2. Математическая модель и её разработка. Формализация задач.
3. Тестирование и отладка программы. Виды ошибок.
4. Алгоритм. Свойства и формы записи алгоритма.
5. Базовые алгоритмические структуры.
6. Язык программирования. Уровни языков программирования. Деление языков высокого уровня.
7. Алфавит и основные определения языка программирования.
8. Программа. Структура программы. Исполняемые и неисполняемые операторы.
9. Определение типа данных.
10. Простые типы данных.
11. Стандартные функции. Таблица стандартных функций.
12. Выражение. Арифметические и логические выражения.
13. Оператор присваивания, пустой и составной операторы.
14. Операторы ввода.
15. Операторы вывода.
16. Условный оператор.
17. Операторы выбора, перехода. Понятие метки.
18. Цикл. Итерационные и вложенные циклы.
19. Операторы цикла.
20. Строковый тип данных.
21. Тип данных - массив. Применение одномерных массивов.
22. Применение многомерных массивов.
23. Определить максимальное значение и его номер в заданном одномерном массиве (Блок-схема, программа, тест).

24. . Определить максимальное значение и его номер в заданном двумерном массиве (Блок-схема, программа, тест).
25. Определение суммы всех элементов заданного одномерного массива. (Блок-схема, программа, тест).
26. Найти наибольший общий делитель двух натуральных чисел (Блок-схема, программа, тест).
27. Сложный тип данных – записи. Оператор присоединения.
28. Подпрограммы. Процедуры и функции.
29. Принципы методов сортировки массива заменой, вставкой, «пузырька». Сортировка массива любым способом.

Г. Самостоятельная работа

Вопросы для контроля самостоятельной работы:

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. Объясните в каком порядке будут выполнены операции

`a = b + c * d << 2 & 8`

`a & 077 != 3`

`a == b || a == c && c < 5`

`c = x != 0`

`a = -1 ++ b -- -5`

`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:

a. Проинициализировать при объявлении;

b. Организовать заполнение массива с клавиатуры;

c. 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 неинициализированными 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:

a. В качестве параметра функции.

b. 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 называется файлом?

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.

Фонд оценочных средств для проведения аттестации уровня сформированности компетенций обучающихся по дисциплине оформляется отдельным документом.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

7.1. Книгообеспеченность

Наименование литературы: автор, название, вид издания, издательство	Год издания	КНИГООБЕСПЕЧЕННОСТЬ	
		Количество экземпляров изданий в библиотеке ВлГУ в соответствии с ФГОС ВО	Наличие в электронной библиотеке ВлГУ
1	2	3	4
Основная литература*			
1. Павлова, О.Н. Основы программирования: лаб. практикум / О.Н. Павлова. Владим. гос.ун-т им. А.Г. и Н.Г. Столетовых. – Владимир: Изд-во ВлГУ	2018	23	+
2. Зоткин, С. П. Программирование на языке высокого уровня С/С++ : конспект лекций / С. П. Зоткин. — 3-е изд. — М. : МИСИ-МГСУ, ЭБС АСВ, 2018. — 140 с. — ISBN 978-5-7264-1810-0. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт].	2018		URL: http://www.iprbookshop.ru/76390.html Режим доступа: для авторизир. пользователей
3. Белева, Л. Ф. Программирование на языке С++ : учебное пособие / Л. Ф. Белева. — Саратов : Ай Пи Эр Медиа, 2018. — 81 с. — ISBN 978-5-4486-0253-5. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт].	2018		URL: http://www.iprbookshop.ru/72466.html Режим доступа: для авторизир. пользователей
Дополнительная литература			
1. Медведева, Ольга Николаевна. Программирование : курс лекций / О. Н. Медведева ; Владимирский государственный университет (ВлГУ)	2011	68	+
2. Зырянов, К. И. Программирование на С++ : учебное пособие / К. И. Зырянов, Н. П. Кисленко. — Новосибирск : Новосибирский государственный архитектурно-строительный университет (Сибстрин), ЭБС АСВ, 2017. — 129 с. — ISBN 978-5-7795-0817-9. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт].	2017		URL: http://www.iprbookshop.ru/85873.html Режим доступа: для авторизир. пользователей
3. Александров, Э. Э. Программирование на языке С в Microsoft Visual Studio 2010 / Э. Э. Александров, В. В. Афонин. — 2-е изд. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 570 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт].	2016		URL: http://www.iprbookshop.ru/73712.html Режим доступа: для авторизир. пользователей

7.2. Периодические издания

Вестник компьютерных и информационных технологий

Вычислительные технологии

Автоматизация. Современные технологии

7.3. Интернет-ресурсы

<https://code-live.ru/tag/cpp-manual/>

<https://proglib.io/p/hiddencpp/>

<http://cppstudio.com/>

<http://www.c-cpp.ru/books/yazyk-s>

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для реализации данной дисциплины имеются специальные помещения для проведения занятий лекционного типа, занятий практического/лабораторного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы. Практические/лабораторные работы проводятся в компьютерных классах кафедры ФиПМ или ИВЦ ВлГУ.

Перечень используемого лицензионного программного обеспечения:

1. Microsoft Windows не ниже 7.0;
2. Microsoft Visual Studio 2010 и выше;
3. Microsoft Office Word 2007 и выше.

Рабочую программу составил ст. преподаватель каф. ФиПМ Павлова О.Н.

(ФИО, подпись)

Рецензент

(представитель работодателя) генеральный директор ООО «ФС Сервис» Квасов Д.С.

(место работы, должность, ФИО, подпись)

Программа рассмотрена и одобрена на заседании кафедры ФиПМ

Протокол № 1 от 31.08.20 года

Заведующий кафедрой _____

(ФИО, подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии направления 02.03.02 Фундаментальная информатика и информационные технологии

Протокол № 1 от 31.08.20 года

Председатель комиссии _____

(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 _____