

**МИНОБРНАУКИ РОССИИ**



Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования

**«Российский государственный гуманитарный университет»**

**(ФГБОУ ВО «РГГУ»)**

**ИНСТИТУТ ИНФОРМАЦИОННЫХ НАУК И ТЕХНОЛОГИЙ БЕЗОПАСНОСТИ**

**Факультет информационных систем и безопасности**

**Кафедра информационных технологий и систем**

**ПРОГРАММИРОВАНИЕ СИ++**

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

Направление подготовки 09.03.03 Прикладная информатика

Направленность (профиль) Прикладная информатика в гуманитарной сфере

Квалификация выпускника - бакалавр

Форма обучения - очная

РПД адаптирована для лиц  
с ограниченными возможностями  
здоровья и инвалидов

Москва 2019

## ПРОГРАММИРОВАНИЕ СИ++

Рабочая программа дисциплины

*Составитель:*

*Кандидат технических наук, доцент, доцент кафедры информационных технологий и систем О.В. Маленкова*

*Ответственный редактор*

*Кандидат технических наук, доцент, зав. кафедрой информационных технологий и систем А.А. Роганов*

УТВЕРЖДЕНО

Протокол заседания кафедры ИТС

№ 12 от 26.06.2019

## ОГЛАВЛЕНИЕ

### Аннотация

### Рабочая программа дисциплины

#### 1. Пояснительная записка

1.1. Цель и задачи дисциплины

1.2. Формируемые компетенции, а также перечень планируемых результатов обучения по дисциплине (знания, умения владения), сформулированные в компетентностном формате

1.3. Место дисциплины в структуре основной образовательной программы

#### 2. Структура дисциплины

#### 3. Содержание дисциплины

#### 4. Информационные и образовательные технологии

#### 5. Фонд оценочных средств для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

5.1. Перечень компетенций с указанием этапов их формирования в процессе освоения дисциплины

5.2. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и опыта деятельности

5.3. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

5.4. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и опыта деятельности

#### 6. Учебно-методическое и информационное обеспечение дисциплины

6.1. Список источников и литературы

6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»,

7. Материально-техническое обеспечение дисциплины Материально-техническое обеспечение дисциплины необходимый для освоения дисциплины

#### 8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов

#### 9. Методические материалы

9.1. Планы практических занятий. Методические указания по организации и проведению

9.2. Методические указания для обучающихся по освоению дисциплины

9.3. Методические рекомендации по подготовке отчетов.

Приложение 1. Аннотация

Приложение 2. Лист изменений на 2020/2021 учебный год

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

### 1. Пояснительная записка

#### 1.1. Цель и задачи дисциплины

*Цель дисциплины:* приобретение знаний, навыков и умений в области технологий и методов программирования, а также освоение основных алгоритмов обработки и хранения больших данных.

*Задачи:*

1. Освоение базовых принципов программирования на языке высокого уровня.
2. Овладение широко применяемыми технологиями и методами программирования на языках C/C++, в том числе специализированными методами обработки и хранения данных.
3. Изучение главных управляющих структур языков C/C++.
4. Формирование навыков и умений по разработке алгоритмов в задачах структурного и объектного-ориентированного программирования.
5. Знакомство со стандартами языков C/C++ и их отличительными особенностями;
6. Изучение процедур отладки и оптимизации программ.
7. Формирование представлений о разработке эффективных приложений и методах оптимизации

#### 1.2. Формируемые компетенции, а также перечень планируемых результатов обучения по дисциплине (знания, умения, владения), сформулированные в компетентностном формате ОПК-7.1; ОПК-7.2; ОПК-7.3

ОПК-7.1 - Знает основные языки программирования, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий.

ОПК- 7.2 - Умеет применять языки программирования, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.

ОПК – 7.3 - Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач.

Дисциплина направлена на формирование следующих компетенций:

Компетенция (код и наименование)	Индикаторы компетенций (код и наименование)	Результаты обучения
ОПК- 7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.1 Знает основные языки программирования, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий.	<p><b>Знать:</b> теоретические основы организации алгоритмов и программ;</p> <p>- операционные системы и оболочки;</p> <p>- современные программные среды разработки информационных систем и технологий.</p> <p><b>Уметь:</b> разрабатывать алгоритмы и программы для решения задач структурного и объектно-ориентированного программирования.</p> <p><b>Владеть:</b> методами оптимизации разрабатываемых алгоритмов по затрачиваемому машинному времени на исполнение</p>

		программы и используемой памяти компьютера.:
	ОПК-7.2 Умеет применять языки программирования, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.	<p><b>Знать:</b> методы разработки эффективных приложений и оптимизации по времени и памяти;</p> <p>- современные программные среды разработки.</p> <p><b>Уметь:</b> применять принципы объектно-ориентированного программирования для построения высокоэффективных приложений.</p> <p><b>Владеть:</b> основными управляющими структурами языков C/C++; методами сортировки и хранения данных; методами и подходами к динамическому выделению памяти в пространстве памяти компьютера методами создания программных прототипов для решения прикладных задач.</p>
	ОПК-7.3 Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач.	<p><b>Знать:</b> главные управляющие структуры языков C/C++;</p> <p>- принципы объектно-ориентированного программирования;</p> <p>- алгоритмы отладки разрабатываемых приложений;</p> <p><b>Уметь:</b> проектировать и разрабатывать информационные прикладное программное обеспечение.</p> <p><b>Владеть:</b> базовыми принципами программирования на языке высокого уровня;</p> <p>- навыками внедрения и адаптации программных ресурсов..</p>

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

*Знать:*

теоретические основы организации алгоритмов; методологию программирования и базовые принципы программирования на языке высокого уровня; главные управляющие структуры языков C/C++; принципы объектно-ориентированного программирования; алгоритмы отладки разрабатываемых приложений; методы разработки эффективных приложений и оптимизации по времени и памяти;

*Уметь:*

применять специализированные методы обработки и хранения данных на языках C/C++; разрабатывать алгоритмы для решения задач структурного и объектно-ориентированного программирования; применять принципы объектно-ориентированного программирования для построения высокоэффективных приложений; порядок и технологию тестирования компонентов программного обеспечения ИС.

*Владеть:*

базовыми принципами программирования на языке высокого уровня; методами оптимизации разрабатываемых алгоритмов по затрачиваемому машинному времени на исполнение программы и используемой памяти компьютера; основными управляющими структурами языков C/C++; методами сортировки и хранения данных; методами и подходами к динамическому выделению памяти в пространстве памяти компьютера.

Программой предусмотрены следующие виды контроля: текущий контроль успеваемости в форме выполнения лабораторных работ и самостоятельных заданий, промежуточный контроль в форме экзамена по билетам.

### **1.3. Место дисциплины в структуре основной образовательной программы**

Дисциплина «Программирование Си ++» является обязательной частью Блока 1 дисциплин учебного плана по направлению подготовки 09.03.03 Прикладная информатика в гуманитарной сфере.

Для освоения дисциплины необходимы компетенции, сформированные в ходе изучения школьных курсов информатики и математики.

В результате освоения дисциплины формируются компетенции, необходимые для изучения следующих дисциплин и прохождения практик: «Информационные системы», «Операционные системы», «Информационно-поисковые системы и машины», «Системы параллельного программирования», «Практика по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности».

**Методами изучения** дисциплины являются лекции с использованием видеоматериалов по темам и практические работы (компьютерный практикум), на которых студенты получают практические навыки программирования.

Общая трудоемкость освоения дисциплины составляет 4 зачетные единицы, 144 часов. Из них: аудиторная работа – 46 часов, самостоятельная работа – 80 часов, контроль 18 часов.

## 2. Структура дисциплины

Общая трудоёмкость дисциплины составляет 4 з.е., 144 ч., в том числе контактная работа обучающихся с преподавателем 46ч., самостоятельная работа обучающихся 80ч.

№ п/ п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоёмкость (в часах)					Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				Лекции	Лабор. зан.	Практ. зан.	Контроль	Самост. работа	
1	Основы программирования на языке С.	2	1-3	4		4		10	Защита отчета по практической работе №1
2	Одномерные и многомерные массивы данных. Динамически выделяемая память на языке С.	2	4-6	4		4		12	Защита отчета по практической работе № 2
3	Методы сортировки массивов. Функциональное программирование.	2	7-	4		6		16	Защита отчета по практической работе №3
4	Программирование классов на языке С++. Концепция объектно-ориентированного программирования.	2	10-12	4		6		16	Защита отчета по практической работе №4
5	Программирование классов на языке С++. Конструктор и деструктор класса. Наследование.	2	13-16	4		6		16	Защита отчета по практической работе № 5
	Промежуточная аттестация	2					18	10	Экзамен по билетам
	<b>Итого</b>			<b>20</b>		<b>26</b>		<b>80</b>	

## 3. Содержание дисциплины

### Тема 1. Основы программирования на языке С.

Основы построения алгоритмов. Способы описания алгоритмов. Структура программы на языке С. Типы данных языка С. Спецификаторы типов данных. Специализированный тип **auto**. Приведение типов. Циклические структуры языка С: **for**, **do...while()**, **while()**. Операторы ветвления **if**, **if()...else**. Оператор выбора **switch()**. Арифметические и логические операции языка С.

**Тема 2. Одномерные и многомерные массивы данных. Динамически выделяемая память на языке C.**

Программирование статических одномерных и многомерных массивов на языке C: инициализация и простейшие действия над ними. Переменные указатели. Динамическое выделение памяти при помощи функций **malloc()**, **calloc()**, **realloc()** и оператора **new**. Очистка памяти операторами **free** и **delete**.

**Тема 3. Методы сортировки массивов. Функциональное программирование.**

Методы сортировки массивов: «пузырьком», «вставками», «метод Шелла», «блочный». Сложность алгоритмов. Программирование пользовательских функций. Передача аргументов по значению и по ссылке. Перегрузка функций. Шаблоны функций. Обработка строковых данных на языке C/C++.

**Тема 4. Программирование классов на языке C++. Концепция объектно-ориентированного программирования.**

Понятие Класс C++. Управление доступом к элементам данных классов. Определение функций-членов класса. Объекты классов Поточный ввод-вывод. Концепция потоков. Операторы **atol()**, **atoi()**, **atof()**. Тип данных **struct**. Объекты типа **struct** и действия над ними. Понятие о реляционной базе данных.

**Тема 5. Программирование классов на языке C++. Конструктор и деструктор класса. Наследование.**

Введение в объектно-ориентированное программирование: инкапсуляцию, полиморфизм, наследование. Понятие о родительском и дочернем классе. Понятие о конструкторе и деструкторе класса. Модификаторы доступа. Процедура наследования. Простое наследование. Доступ к наследуемым компонентам. Особенности наследования классов. Множественное наследование. Виртуальное наследование. Адреса базовых классов.



#### 4. Информационные и образовательные технологии

При реализации рабочей программы дисциплины «Программирование СИ++» используются следующие информационные и образовательные технологии:

№ п/п	Наименование раздела	Виды учебных занятий	Формируемые компетенции	Образовательные технологии
1	2	3	4	5
1	Основы программирования на языке С.	Лекции 1.  Практическая работа 1  Самостоятельная работа	7.1, 7.2, 7.3	Лекция с применением проектора  Консультирование по вопросам практических работ Прием отчета по практической работе №1 Прием самостоятельной работы
2	Одномерные и многомерные массивы данных. Динамически выделяемая память на языке С.	Лекции 2.  Практическая работа 2  Самостоятельная работа	7.2	Лекция с применением проектора Консультирование по вопросам практических работ Прием отчета по практической работе № 2 Прием самостоятельной работы
3	Методы сортировки массивов. Функциональное программирование.	Лекции 3. Лабораторная работа 3  Самостоятельная работа	7.3	Лекция с применением проектора Консультирование по вопросам практических работ Прием отчета по практической работе № 3 Прием самостоятельной работы
4	Программирование классов на языке С++. Концепция объектно-ориентированного программирования.	Лекции 4.  Лабораторная работа 4  Самостоятельная работа	7.2, 7.3	Лекция с применением проектора Консультирование по вопросам практических работ Прием отчета по практической работе № 4 Прием самостоятельной работы
5	Программирование классов на языке С++. Конструктор и деструктор класса. Наследование.	Лекции 5  Лабораторная работа 5  Самостоятельная работа	7.2, 7.3	Лекция с применением проектора Консультирование по вопросам практических работ Прием отчета по практической работе № 5 Прием самостоятельной работы

## 5. Фонд оценочных средств для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

### 5.1. Перечень компетенций с указанием этапов их формирования в процессе освоения дисциплины

№ п/п	Контролируемые разделы дисциплины	Код контролируемой компетенции (компетенций)	Наименование оценочного средства
1	Основы программирования на языке C.	7.1, 7.2, 7.3	Планы лабораторных работ. Тема 1. Задание на работу 1, контрольные вопросы
2	Одномерные и многомерные массивы данных. Динамически выделяемая память на языке C.	7.2	Планы лабораторных работ. Тема 2. Задание на работу 2, контрольные вопросы
3	Методы сортировки массивов. Функциональное программирование.	7.3	Планы лабораторных работ. Тема 3. Задание на работу 3, контрольные вопросы
4	Программирование классов на языке C++. Концепция объектно-ориентированного программирования.	7.2, 7.3	Планы лабораторных работ. Тема 4. Задание на работу 4, контрольные вопросы
5	Программирование классов на языке C++. Конструктор и деструктор класса. Наследование.	7.2, 7.3	Планы лабораторных работ. Тема 5. Задание на работу 5, контрольные вопросы

### 5.2. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Форма контроля	Срок отчетности	Макс. количество баллов	
		За одну работу	Всего
Текущий контроль: Защита практических работ	3, 6, 9, 12, 15 недели		60 баллов
Практическая работа 1		5	45 баллов
Практическая работа 2		10	
Практическая работа 3		10	
Практическая работа 4		10	
Практическая работа 5		10	
Самостоятельная работа		5	15 баллов
Промежуточная аттестация (экзамен)	сессия		40 баллов
Итого за семестр (дисциплину)			100 баллов

Полученный совокупный результат (максимум 100 баллов) конвертируется в традиционную шкалу оценок и в шкалу оценок Европейской системы переноса и накопления кредитов (European Credit Transfer System; далее – ECTS) в соответствии с таблицей:

100-балльная шкала	Традиционная шкала		Шкала ECTS
95 – 100	отлично	зачтено	A
83 – 94			B
68 – 82	хорошо		C
56 – 67	удовлетворительно		D
50 – 55			E
20 – 49	неудовлетворительно	не зачтено	FX
0 – 19			F

### 5.3. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Критерии оценивания / Уровень требований к обучающемуся	Макс. кол-во баллов
<b>Текущий контроль, всего в т.ч.:</b>	<b>60</b>
<b>Практическая работа</b>	<b>5 -10</b>
Задания выполнены не полностью и (или) допущены две и более ошибки или три и более недочета	1-5
Задания выполнены полностью, но допущены два-три недочета, в т. ч. при ответе на контрольные вопросы	6-7
Задания выполнены полностью, возможна одна неточность, ответы на контрольные вопросы правильные	8-10
<b>Промежуточная аттестация, всего в т.ч.:</b>	<b>40</b>
<b>Ответ на вопросы билета (2 вопроса – по 20 баллов максимум каждый)</b>	<b>20</b>
Теоретическое содержание <i>почти не освоено</i> : - фрагментарные знания материала, наличие грубых ошибок в ответе.	1-5
Теоретическое содержание <i>освоено частично</i> : - демонстрируется не всегда осознанное воспроизведение программного материала, доказательство теорем проводится с ошибками или фрагментарно; - допущено не более двух-трех недочетов.	6-9
Теоретическое содержание освоено <i>почти полностью</i> : - ответ удовлетворяет основным требованиям: знание математических фактов и закономерностей, использование этих знаний при обосновании утверждений и теорем; - допущено не более одного-двух недочетов, но обучающийся может их исправить самостоятельно или с небольшой помощью преподавателя.	10-14
Теоретическое содержание освоено <i>полностью</i> : - доказательства приведены с требуемым обоснованием, использована математическая терминология, рисунки и графики, сопутствующие ответу, выполнены верно; - ответ строится по собственному плану, установлена связь с материалом, усвоенным при изучении других дисциплин.	15-20

#### 5.4. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности

##### Текущий контроль

##### *Самостоятельная работа.*

###### Самостоятельная работа 1

Набрать текст программы, представленный ниже. Проанализировать значения переменных после каждой операции присваивания. Проверить порядок выполнения операций в каждом выражении, содержащем несколько операций присваивания, разделив каждый оператор-выражение на несколько операторов, выполняемых последовательно. В функциях ввода и вывода изменить спецификаторы формата, проанализировать полученные результаты.

```
#include <stdio.h>
#include <stdlib.h>
int main (void)
int a, b = 5, c;
double x, y = -.5, z;
printf("a=");
scanf("%d", &a);
x = c = a;
printf("x = c = a : a=%d c=%d x=%f\n",a, c, x);
a += b;
printf("a += b : a=%d\n", a);
x *= b+a;
printf("x *= b+a : x=%lf\n", x);
b += a--;
printf("b += a-- : a=%d b=%d\n", a, b);
x -= ++c;
printf("x -= ++c : c=%d x=%lf\n", c, x);
c = a/b;
printf("c = a/b : c=%4d\n",c);
c = a%b;
printf("c = a%%b : c=%d\n",c);
y += (a+1)/a++;
printf("y += (a+1)/a++ : a=%d y=%.3lf\ty=%.0lf\n",a, y, y);
b = 3*(y-.6)+2*b+1;
printf("b = 3*(y-.6)+2*b+1 : b=%d y=%.1lf\n",b, y);
z = a/2;
printf("z = a/2 : z = a/2 : z=%lf\n", z);
z = (double)a/2;
printf("z = (double)a/2 : z=%lf\n", z);
y = (x = 5.7)/2;
printf("y = (x = 5.7)/2 : x=%lf y=%lf\n", x, y);
y = (int)x/2;
printf("y = (int)x/2 : y=%f\n", y);
z = (b-3)/2 - x/5 +(c/=2) + 1/4*z - y++ + ++b/3.;
printf("z = (b-3)/2 - x/5 +(c/=2) + 1/4*z - y++ \+ ++b/3. :\n\ta=%d b=%d c=%d x=%lf y=%lf\n",a,b,c,x,y,z);
system("pause");
return 0;
```

###### Самостоятельная работа 2

Написать программу для вычисления значений следующих выражений:

a=5, c=5

a=a+b-2

c=c+1, d=c-a+d

a=a\*c, c=c-1

a=a/10, c=c/2, b=b-1, d=d\*(c+b+a)

Выражения, записанные в одной строке, записывать одним оператором выражением. Переменные c и d объявить как целые, переменные a и b – как вещественные. Значения переменных b и d вводить с клавиатуры. После вычисления каждого выражения выводить на экран значения всех переменных.

Самостоятельная работа 3.

Набрать текст программы, представленный ниже. Проанализировать выдаваемые программой результаты. Объяснить, почему они именно такие.

```
#include <stdlib.h>
#include <stdio.h>
#include <limits.h>
#include <float.h>
main()
char c;
unsigned char uc;
int i;
unsigned u;
short s;
long l;
float f;
double d;
printf("sizeof(c)=%d\\tsizeof(uc)=%d\\nsizeof(i)\\tsizeof(u)=%d\\tsizeof(s)=%d\\tsizeof(l)=%d\\n\\tsizeof(f)=%d\\tsizeof(d)=%d\\n\\n", sizeof(c), sizeof(uc), sizeof(i), sizeof(u), sizeof(s), sizeof(l), sizeof(f), sizeof(d));
uc=c=CHAR_MAX;
printf("CHAR_MAX : c=%d uc=%d\\n", c, uc);
c++; uc++;
printf("CHAR_MAX+1 : c=%d uc=%d\\n", c, uc);
uc=c=CHAR_MIN;
printf("CHAR_MIN : c=%d uc=%d\\n", c, uc);
c=uc=UCHAR_MAX;
printf("UCHAR_MAX : c=%d uc=%d\\n", c, uc);
c++; uc++;
printf("UCHAR_MAX+1 : c=%d uc=%d\\n", c, uc);
uc=c=-5;
printf("-5 : c=%d uc=%d\\n", c, uc);
c=-5; uc=5;
printf("char and unsigned char -5>5 : %d\\n\\n", c>uc);
c=s=SHRT_MAX;
uc=s;
printf("SHRT_MAX : c=%d uc=%d s=%d\\n", c, uc, s);
s++;
printf("SHRT_MAX+1 : s=%d\\n", s);
c=s; uc=s;
printf("%d : c=%d uc=%d\\n", SHRT_MIN, c, uc);
s=0; c=s; uc=s;
```

```

printf("0 : c=%d uc=%d s=%d\n", c, uc, s);
i=INT_MAX;
l=i; u=i;
printf("INT_MAX : i=%d u=%u l=%ld\n", i, u, l);
i++; l++; u++;
printf("INT_MAX+1 : i=%d u=%u l=%ld\n", i, u, l);
i=INT_MIN;
l=i; u=i;
printf("INT_MIN : i=%d u=%u l=%ld\n", i, u, l);
u=UINT_MAX;
i=u; l=u;
printf("UINT_MAX : i=%d u=%u l=%ld\n", i, u, l);
u=i-5;
printf("-5 : i=%d u=%u\n", i, u);
i=-5; u=5;
printf("int and unsigned int -5>5 : %d\n", i>u);
c=-5; u=5;
printf("char and unsigned int -5>5 : %d\n",
c>u);
i=5.1;
printf("i=5.1 : i=%d\n", i);
i=5.9;
printf("i=5.9 : i=%d\n", i);
d=f=FLT_MAX;
printf("FLT_MAX : f=%g d=%g\n", f, d);
d=f=FLT_MIN;
printf("FLT_MIN : f=%g d=%g\n", f, d);
d=f=FLT_EPSILON;
printf("FLT_EPSILON : f=%g d=%g\n", f, d);
f=1e10;
printf("1e10 : f=%f\n", f);
f=1e11;
printf("1e11 : f=%f\n", f);
f=1234567890;
printf("1234567890 : f=%f\n", f);
d=DBL_MAX;
printf("DBL_MAX : d=%g\n", d);
d=DBL_MIN;
printf("DBL_MIN : d=%g\n", d);
d=DBL_EPSILON;
printf("DBL_EPSILON : d=%g\n", d);
d=1e15+1;
printf("1e15+1 : d=%lf\n", d);
d=1e16+1;
printf("1e16+1 : d=%lf\n", d);
f=10000*100000;
f+=1;
f-=4*250000000;
printf("1 : f=%f\n", f);
f=10000*100000+1-4*250000000;
printf("1 : f=%f\n", f);
d=10000*100000+1-4*250000000;

```

```
printf("1 : d=%lf\n", d);
d=1e20*1e20+1000-1e22*1e18;
printf("1000 : d=%lf\n", d);
system("pause");
return 0;
```

### Промежуточная аттестация

#### *Примерные контрольные вопросы по курсу*

1. Алгоритмы: определение, свойства. (ОПК-7)
2. Жизненный цикл программного обеспечения. (ОПК-7.1)
3. Обзор языков программирования. (ОПК-7.1)
4. Способы описания алгоритмов. (ОПК-7.1)
5. Структура программы на языке Си. (ОПК-7.1)
6. Типы данных в языке Си. Определение простых переменных в программе. (ОПК-7.2)
7. Арифметические операции языка Си. (ОПК-7.2)
8. Логические операции языка Си (ОПК-7.2)
9. Операторы языка Си: условный, оператор-переключатель. (ОПК-7.2)
10. Операторы циклов в языке Си. (ОПК-7.2)
11. Одномерные массивы в Си: определение, способы инициализации. (ОПК-7.2)
12. Типовые операции с одномерными массивами: поиск минимума, сумма элементов. (ОПК-7.2)
13. Двумерные массивы в языке Си: определение, способы инициализации. (ОПК-7.2)
14. Указатели: назначение, определение, инициализация. (ОК-7.2)
15. Динамическое выделение памяти. (ОПК-7.2)
16. Функции ввода-вывода в C++. (ОПК-7.2)
17. Универсальный метод выделения динамической памяти под двумерные массивы. (ОК-7, ПК-2)
18. Строки в языке Си: определение, инициализация, кодирование символов. (ОК-7, ПК-2)
19. Функции для работы со строками языка Си. (ОК-7, ПК-2)
20. Структурный тип данных: определение типа и переменных, инициализация структур. (ОПК-7.2)
21. Вложенные структуры, массивы структур. Динамическое выделение памяти под массивы структур. (ОПК-7.3)
22. Перечислимый тип данных: определение, назначение, пример использования. (ОПК-7.3)
23. Динамические списки: назначение, основные принципы организации. (ОПК-7.1)
24. Односвязный список: определение, особенности использования, типовые алгоритмы обработки. (ОПК-7.1)
25. Сравнительный анализ односвязного списка и массива структур. (ОПК-7.2)
26. Двусвязный список: определение, особенности использования, типовые алгоритмы обработки. (ОПК-7.2)
27. Поточковый ввод-вывод. Определение потока, стандартные потоки, типы потоков. (ОПК-7.2)
28. Основные функции контроля состояния потоков. (ОПК-7.2)
29. Основные функции ввода строковой информации в поток и вывода информации из потока. (ОПК-7.2)
30. Форматированный потоковый ввод-вывод в языке Си. (ОПК-7.2)
31. Функции в Си: определение, прототип, вызов. (ОПК-7.1)

32. Основные преимущества структурного стиля программирования. (ОПК-7.1)
33. Формальные и фактические параметры. Связь формальных и фактических параметров функций. (ОПК-7.2)
34. Порядок вызова функции и возврата из нее. Передача параметров по значению и по адресу. (ОПК-7.3)
35. Функции и массивы: массивы как параметры функций. (ОПК-7.3)
36. Функции с умалчиваемыми значениями параметров. (ОПК-7.2)
37. Передача параметров функции main. (ОПК-7.2)
38. Ссылки: назначение, определение, использование. (ОПК-7.2)
39. Рекурсивные вызовы функций: назначение, примеры рекурсивных задач, прямая и косвенная рекурсия. (ОПК-7.2)
40. Разворачивание и свертывание рекурсии в памяти, глубина рекурсии. (ОПК-7.3)
41. Алгоритм быстрой обменной сортировки, его рекурсивная реализация (ОПК-7.3)
42. Указатели на функции: определение, назначение, примеры использования. (ОПК-7.3)
43. Указатели на функции как параметры функций. (ОПК-7.3)
44. Перегрузка функций: определение, достоинства, ограничения. Примеры реализации. (ОПК-7.2)
45. Препроцессор: директивы включения файлов и замены идентификаторов. (ОПК-7.3)
46. Препроцессор: директивы условной компиляции. (ОПК-7.3)
47. Принципы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм. (ОПК-7.3)
48. Определение классов и объектов в программах на языке Си++ (ОПК-7.3)
49. Область видимости компонент класса. (ОПК-7.3)
50. Правила наследования классов в языке Си++. (ОПК-7.3)
51. Библиотеки `cstdlib`, `conio`, `iostream`, `math`, `algorithm`, `vector`, `cstdio`. (ОПК-7.3)
52. Операции ++ и --. Префиксное и постфиксное использование операций. Применение к указателям. (ОПК-7.3)



## 6. Учебно-методическое и информационное обеспечение дисциплины

### 6.1. Список источников и литературы

#### Литература

##### Основная

1. Бедердинова, О. И. Программирование на языках высокого уровня: учеб. пособие / О.И. Бедердинова, Т.А. Минеева, Ю.А. Водовозова. — Москва: ИНФРА-М, 2019. 159 с. - Текст: электронный. - URL: <https://new.znaniium.com/catalog/product/1044396>.
2. Корнеев, В. И. Программирование графики на C++. Теория и примеры: учеб. пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. — Москва: ИД «ФОРУМ»: ИНФРА-М, 2019. - 517 с. URL: <https://new.znaniium.com/catalog/product/1018909>.
3. Воронцова, Е. А. Программирование на C++ с погружением: практические задания и примеры кода - Москва: НИЦ ИНФРА-М, 2016. - 80 с. Текст: электронный. - URL: <https://new.znaniium.com/catalog/product/563294>.

##### Дополнительная

1. Литвиненко, В. А. Программирование на C++ задач на графах: Учебное пособие / Литвиненко В.А. - Таганрог: Южный федеральный университет, 2016. - 83 с. URL: <https://new.znaniium.com/catalog/product/997083>.
2. Русанова, Я. М. C++ как второй язык в обучении приемам и технологиям программирования: учеб. пособие / Я. М. Русанова. - Ростов-на-Дону: Издательство ЮФУ, 2010. - 200 с. - - URL: <https://znaniium.com/catalog/product/550811>.
3. Дорогов, В.Г. Основы программирования на языке C: учеб. пособие / В.Г. Дорогов, Е.Г. Дорогова; под ред. проф. Л.Г. Гагариной. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 224 с. - URL: <https://new.znaniium.com/catalog/product/1016471>.

### 6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимый для освоения дисциплины

1. Форум программистов и сисадминов Киберфорум <http://www.cyberforum.ru/>
2. Клуб программистов <https://programmersforum.ru/>
3. Форум программистов <https://programmersforum.ru/>

### 6.3 Перечень современных профессиональных баз данных (БД) и информационно-справочных систем (ИСС)

№п/п	Наименование
1	Международные реферативные наукометрические БД, доступные в рамках национальной подписки в 2019 г. Scopus
2	Компьютерные справочные правовые системы Консультант Плюс, Гарант

## **7. Материально-техническое обеспечение дисциплины**

Для материально-технического обеспечения дисциплины «Программирование C++»

- компьютерный класс

Компьютерный класс – ауд. № 114: 1 компьютер преподавателя, 9 компьютеров обучающихся, маркерная доска, проектор.

- предустановленное программное обеспечение:

- 1) Операционная система Windows 10 (лицензия 68526624.
- 2) Microsoft office 2010 Pro (лицензия 49420326 от 08.12.2011)
- 3) Microsoft SQL Server 2008 (лицензия 46931055 от 20.05.2010)
- 4) Microsoft Visual Professional 2019 (лицензия 63202190)

## **8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов**

## **9. Методические материалы**

### **9.1. Планы практических занятий.**

#### **Тема 1. Основы программирования на языке C.**

**Цель занятия:** закрепление и формирование знаний, умений, навыков

Форма проведения – рассмотрение основных структур и алгоритмов

#### ***Практическая работа 1:***

1. Согласно заранее определенному варианту выполнить программирование операторов цикла и ветвления для решения алгебраической задачи;
2. Типизацию переменных задачи осуществить согласно принятым стандартам языков C/C++;
3. Подготовить развернутый отчет о проделанной работе.

*Контрольные вопросы:*

1. Структура программы на языке C.
2. Типы данных языка C. Приведение типов.
3. Арифметические и логические операции языка C.
4. Циклические структуры языка C: **for**, **do...while()**, **while()**.
5. Операторы ветвления **if**, **if()...else**.
6. Оператор выбора **switch()**.

#### **Тема 2. Одномерные и многомерные массивы данных. Динамически выделяемая память на языке C.**

**Цель занятия:** закрепление и формирования знаний, умений, навыков

Форма проведения – рассмотрение основных структур и алгоритмов

#### ***Практическая работа 2:***

1. Согласно заранее определенному варианту выполнить объявление многомерного статического массива;
2. Используя функции динамического выделения памяти, выделить для динамически память для прежде определенного массива. Выполнить поиск и перестановку минимального и максимального элемента.
3. Подготовить развернутый отчет о проделанной работе.

*Контрольные вопросы:*

1. Программирование статических одномерных и многомерных массивов на языке C: инициализация и простейшие действия над ними.
2. Переменные-указатели.
3. Динамическое выделение памяти при помощи функций **malloc()**, **calloc()**, **realloc()** и оператора **new**.
4. Очистка памяти операторами **free** и **delete**.

**Тема 3. Методы сортировки массивов. Функциональное программирование.**

**Цель занятия:** закрепление и формирования знаний, умений, навыков

Форма проведения – рассмотрение основных структур и алгоритмов

***Практическая работа 3:***

1. Согласно заранее определенному варианту выполнить программирование алгоритмов сортировки с использованием прежде изученных операторов цикла и ветвления;
2. Методы сортировки реализуются посредством пользовательских функций;
3. Подготовить развернутый отчет о проделанной работе.

*Контрольные вопросы:*

1. Методы сортировки массивов: «пузырьком», «вставками», «метод Шелла», «блочный».
2. Сложность алгоритмов.
3. Программирование пользовательских функций.
4. Передача аргументов по значению и по ссылке.
5. Перегрузка функций.
5. Шаблоны функций.

**Тема 4. Программирование классов на языке C++. Концепция объектно-ориентированного программирования.**

**Цель занятия:** закрепление и формирования знаний, умений, навыков

Форма проведения – рассмотрение основных структур и алгоритмов

***Практическая работа 4:***

1. Согласно заранее определенному варианту выполнить обработку C-подобных строковых данных и осуществить чтение/запись в файл;
2. Предусмотреть обработку текстовых(числовых) данных, хранящихся в файле, при помощи инструмента **struct**;
3. Подготовить развернутый отчет о проделанной работе.

*Контрольные вопросы:*

1. Обработка строковых данных на языке C/C++.
2. Поточковый ввод-вывод.
3. Операторы **atol()**, **atoi()**, **atof()**.
4. Тип данных **struct**. Объекты типа **struct** и действия над ними.
5. Понятие о реляционной базе данных.

## Тема 5. Программирование классов на языке C++. Конструктор и деструктор класса. Наследование.

**Цель занятия:** закрепление и формирования знаний, умений, навыков

Форма проведения – рассмотрение основных структур и алгоритмов

### Лабораторная работа 5:

1. Согласно заранее определенному варианту и используя принципы объектно-ориентированного программирования организовать создание объекта типа **class** в состав, которого входят методы сортировки динамически инициализированного двумерного массива, а также определены конструктор и деструктор класса;
2. Предусмотреть возможность создания дочерних классов на базе созданного родительского класса;
3. Подготовить развернутый отчет о проделанной работе.

### Контрольные вопросы:

1. Введение в объектно-ориентированное программирование: инкапсуляцию, полиморфизм, наследование.
2. Класс C++. Понятие о конструкторе и деструкторе класса.
3. Модификаторы доступа.
4. Понятие о родительском и дочернем классе.
5. Процедура наследования.

## 9.2. Методические указания для обучающихся по освоению дисциплины

Вид работы	Содержание (перечень вопросов)	Трудоемкость самостоятельной работы (в часах)	Рекомендации
Тема №1. Основы программирования на языке C.			
Подготовка к практической работе	Тему практической работы и контрольные вопросы см. в разделе 9.1	10	См. Список литературы, Интернет-ресурсов в разделе 6.
Тема № 2. Одномерные и многомерные массивы данных. Динамически выделяемая память на языке C.			
Подготовка к практической работе	Тему практической работы и контрольные вопросы см. в разделе 7.1	12	См. Список литературы, Интернет-ресурсов в разделе 6.
Тема № 3. Методы сортировки массивов. Функциональное программирование.			
Подготовка к практической работе	Тему практической работы и контрольные вопросы см. в разделе 7.1	16	См. Список литературы, Интернет-ресурсов в разделе 6.
Тема № 4. Программирование классов на языке C++. Концепция объектно-ориентированного программирования.			
Подготовка к практической работе	Тему практической работы и контрольные вопросы см. в разделе 7.1	16	См. Список литературы, Интернет-ресурсов в разделе 6.
Тема № 5. Программирование классов на языке C++. Конструктор и деструктор класса. Наследование.			
Подготовка к практической работе	Тему практической работы и контрольные вопросы см. в разделе 7.1	16	См. Список литературы, Интернет-ресурсов в разделе 6.
Промежуточная аттестация			

<b>Подготовка к экзамену</b>	См. Содержание дисциплины (раздел 3)	10	См. Примерные контрольные вопросы по курсу в разделе 5.4, Список литературы, Интернет-ресурсов в разделе 6
<b>Итого по дисциплине</b>		<b>80</b>	

### **Методические рекомендации по подготовке отчетов.**

Письменными работами по данной дисциплине являются отчеты о практических работах, которые обучающиеся выполняют и оформляют в соответствии с требованиями.

В среде приложения MS Office Word набирается текст по следующему шаблону. По центру пишется «Отчет о практической работе №\_\_ «Наименование Практической работы» (в соответствии с наименованием, указанным в методических указаниях), указывается фамилия, имя и отчество студента, а также шифр его группы.

Затем в отчете указывается наименование раздела «1. Основы программирования на языке C» и под наименованием раздела формулируется цель соответствующей работы. Точно также выполняется и оформляется раздел «2. Методика работы». Разделы «3.1-3.4» выполняются и оформляются в отчете в соответствии с составом и содержанием выполненной практической работы. В тексте соответствующего раздела, например, см. раздел 3.3, необходимо поместить скриншоты результатов практической работы, например, условие задачи, исходные данные, листинг программ, таблиц и др. Скриншоты снабжаются наименованиями, например, «Рис.1. Алгоритм решения задачи», пишется наименование под скриншотом (рисунком). При наличии в отчете таблицы над ней пишется наименование, например, «Таблица 1. Дефекты программы». Шрифт текста отчета Times New Roman, размер шрифта – 14, абзацный отступ - 1 см., выравнивание «по ширине», интервал между строк – 1,5. Наименование разделов выделять жирным шрифтом. Данный раздел 3.5. в отчете не оформляется. Он рассматривается как методические рекомендации для данной и всех последующих практических работ.

В конце каждого отчета оформляется раздел «Выводы». По каждому из этапов приводятся краткие выводы (резюме) по методике, средствам, ресурсам, которые можно использовать в решении рассматриваемой задачи.

## АННОТАЦИЯ

Дисциплина «Программирование СИ ++» является обязательной частью Блока 1 дисциплин учебного плана по направлению подготовки 09.03.03 Прикладная информатика в гуманитарной сфере. Дисциплина реализуется на факультете информационных систем и безопасности кафедрой информационных технологий и систем.

*Цель дисциплины:* приобретение знаний, навыков и умений в области технологий и методов программирования, а также освоение основных алгоритмов обработки и хранения данных.

*Задачи:*

1. Освоение базовых принципов программирования на языке высокого уровня.
2. Овладение широко применяемыми технологиями и методами программирования на языках C/C++, в том числе специализированными методами обработки и хранения данных.
3. Изучение главных управляющих структур языков C/C++.
4. Формирование навыков и умений по разработке алгоритмов в задачах структурного и объектно-ориентированного программирования.
5. Знакомство со стандартами языков C/C++ и их отличительными особенностями;
6. Изучение процедур отладки и оптимизации программ.
7. Формирование представлений о разработке эффективных приложений и методах оптимизации.

Дисциплина направлена на формирование следующих компетенций: ОПК-7.1; ОПК-7.2; ОПК-7.3

В результате освоения дисциплины обучающийся должен:

*Знать:* базовые принципы программирования на языке высокого уровня; главные управляющие структуры языков C/C++; принципы объектно-ориентированного программирования; алгоритмы отладки разрабатываемых приложений; методы разработки эффективных приложений и оптимизации по времени и памяти;

*Уметь:* применять специализированные методы обработки и хранения данных на языках C/C++; разрабатывать алгоритмы для решения задач структурного и объектно-ориентированного программирования; применять принципы объектно-ориентированного программирования для построения высокоэффективных приложений.

*Владеть:* базовыми принципами программирования на языке высокого уровня; методами оптимизации разрабатываемых алгоритмов по затрачиваемому машинному времени на исполнение программы и используемой памяти компьютера; основными управляющими структурами языков C/C++; методами сортировки и хранения данных; методами и подходами к динамическому выделению памяти в пространстве памяти компьютера.

Рабочей программой предусмотрены следующие виды контроля: текущий контроль успеваемости в форме защиты практических заданий, промежуточная аттестация в форме экзамена.

Общая трудоемкость освоения дисциплины составляет 4 зачетные единицы, 144 часа.

УТВЕРЖДЕНО  
 Протокол заседания кафедры  
 № 12 от 08.06.2020

### ЛИСТ ИЗМЕНЕНИЙ 1

в рабочей программе дисциплины - Программирование (C++)  
 по направлению подготовки - «Прикладная информатика»,  
 направленность «Прикладная информатика в гуманитарной сфере»  
 на 2020/2021 учебный год

#### 1. Добавлена основная литература раздел 6.1.

Огнева М.В. Программирование на языке C++: практический курс: учебное пособие для вузов/ М.В. Огнева, Е.В. Кудрина. – Москва: Издательство Юрайт, 2020. – 335 с. URL: <https://urait.ru/viewer/programmirovanie-na-yazyke-s-prakticheskiy-kurs-454165#page/2>

#### 2. Перечень программного обеспечения (ПО)

№п /п	Наименование ПО	Производитель	Способ распространения
1	Microsoft Office 2010 Pro	Microsoft	лицензионное
2	Microsoft Visual Professional 2019	Microsoft	лицензионное
3	Windows 10	Microsoft	лицензионное
4	Mozilla Firefox 52.8.1 ESR	Mozilla	свободный доступ
5	Платформа ZOOM	Zoom	лицензионное

#### 3. Перечень современных профессиональных баз данных (БД) и информационно-справочные систем (ИСС)

№п/п	Наименование
1	Международные реферативные наукометрические БД, доступные в рамках национальной подписки в 2020 г. Scopus
2	Компьютерные справочные правовые системы Консультант Плюс, Гарант