

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**



Федеральное государственное бюджетное образовательное учреждение
высшего 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 (C++)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

38.03.01 Экономика

Код и наименование направления подготовки/специальности

Экономика и анализ данных

Наименование направленности (профиля)/ специализации

Уровень высшего образования: *бакалавриат*

Форма обучения: *Очно-заочная*

РПД адаптирована для лиц
с ограниченными возможностями
здоровья и инвалидов

Москва 2024

ПРОГРАММИРОВАНИЕ (C++)

Рабочая программа дисциплины

Составитель:

к.с.-х.н., доцент, заведующая кафедрой Информационных технологий и систем
Н.Ш. Шукенбаева

УТВЕРЖДЕНО:

Протокол заседания кафедры

№ 3 от 28 марта 2024 года

ОГЛАВЛЕНИЕ

1	Пояснительная записка.....	4
1.1	Цель и задачи дисциплины.....	4
1.2	Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций.....	4
1.3	Место дисциплины в структуре основной образовательной программы	5
2	Структура дисциплины.....	7
3.	Содержание дисциплины.....	7
4.	Образовательные технологии.....	8
5.	Оценка планируемых результатов обучения.....	8
5.2	Критерии выставления оценки по дисциплине	9
5.3	Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине	11
6.	Учебно-методическое и информационное обеспечение дисциплины	16
6.1	Список источников и литературы	16
6.2.	Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимый для освоения дисциплины	16
6.3	Профессиональные базы данных и информационно-справочные системы	16
8.	Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов.....	17
9.	Методические материалы	18
9.1.	Планы практических занятий	18
9.2.	Методические рекомендации для обучающихся по освоению дисциплины	20
9.3	Методические рекомендации по подготовке по подготовке письменных работ.....	21
	Приложение 1.....	22

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 Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Изучение дисциплины направлено на формирование у обучающихся следующих компетенций:

Компетенция (код и наименование)	Индикаторы компетенций (код и наименование)	Результаты обучения
ОПК-5 Способен использовать современные информационные технологии и программные средства при решении профессиональных задач; инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.1. Выбирает наиболее эффективные информационные технологии и программные средства для решения профессиональных задач	Знать: современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности; Уметь: выбирать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.; Владеть: навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности..
	ОПК-5.2. Использует современные информационные технологии и программные средства для решения профессиональных задач	Знать: теоретические основы организации алгоритмов и программ; операционные системы и оболочки; современные программные среды разработки информационных систем и технологий; Уметь: разрабатывать алгоритмы и программы для решения профессиональных задач; Владеть: навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач.

	ОПК-5.3. Принимает участие инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем	Знать: основные языки программирования и современные программные среды разработки информационных систем и технологий.; Уметь: устанавливать программное и аппаратное обеспечение информационных и автоматизированных систем; Владеть: навыками инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем.
ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1. Знает основные языки программирования, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий	Знать: теоретические основы организации алгоритмов, структур данных и программ; современные программные среды разработки. Уметь: разрабатывать алгоритмы и программы для решения задач. Владеть: методами оптимизации разрабатываемых алгоритмов по затрачиваемому машинному времени на исполнение программы и используемой памяти компьютера.:
	ОПК-8.2. Умеет применять языки программирования, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов и решения других прикладных задач в профессиональной деятельности	Знать: методы разработки эффективных приложений и оптимизации по времени и памяти; современные программные среды разработки. Уметь: применять принципы программирования для построения высокоэффективных приложений. Владеть: основными управляющими структурами языков программирования; алгоритмами и структурами данных; методами и подходами к динамическому выделению памяти в пространстве памяти компьютера методами создания программных прототипов для решения прикладных задач .
	ОПК-8.3. Владеет навыками программирования для решения задач в профессиональной сфере	Знать: главные управляющие структуры языков программирования; принципы программирования; алгоритмы отладки разрабатываемых приложений; Уметь: проектировать и разрабатывать информационные прикладное программное обеспечение. Владеть: базовыми принципами программирования на языке высокого уровня; навыками внедрения и адаптации программных ресурсов..

1.3 Место дисциплины в структуре основной образовательной программы

Дисциплина «Программирование (C++)» относится к обязательной части блока дисциплин учебного плана.

Для освоения дисциплины необходимы компетенции, сформированные в ходе изучения дисциплины «Алгоритмы и структуры данных».

В результате освоения дисциплины формируются компетенции, необходимые для изучения следующих дисциплин и прохождения практик: «Программирование (Python)», «Базы данных», «Моделирование систем и методы оптимизации».

2 Структура дисциплины

Общая трудоёмкость дисциплины составляет 4 з.е., 144 академических часов

Структура дисциплины для очной формы обучения

Объем дисциплины в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Семестр	Тип учебных занятий	Количество часов
3	Лекции	14
3	Практические работы	18
Всего:		32

Объем дисциплины (модуля) в форме самостоятельной работы обучающихся составляет 94 академических часов, промежуточная аттестация 18 ч.

3. Содержание дисциплины

Тема 1. Основы программирования на языке C.

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

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 Поточковый 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. Образовательные технологии

Для проведения учебных занятий по дисциплине используются различные образовательные технологии. Для организации учебного процесса может быть использовано электронное обучение и (или) дистанционные образовательные технологии.

№ п/п	Наименование раздела	Виды учебных занятий	Образовательные технологии
1	2	3	4
1.	Тема 1. Основы программирования на языке C.	Лекция 1. Практическая работа 1 Самостоятельная работа	Интерактивная лекция Занятия с использованием специализированного ПО Изучение материала по теме
2.	Тема 2. Одномерные и многомерные массивы данных Динамически выделяемая память на языке C.	Лекция 2. Практическая работа 2 Самостоятельная работа	Интерактивная лекция Занятия с использованием специализированного ПО Подготовка к занятию с использованием ЭБС
3.	Тема 3. Методы сортировки массивов. Функциональное программирование.	Лекция 3. Практическая работа 3 Самостоятельная работа	Интерактивная лекция Занятия с использованием специализированного ПО Подготовка к занятию с использованием ЭБС
4.	Тема 4. Программирование классов на языке C++. Концепция объектно-ориентированного программирования.	Лекция 4. Практическая работа 4 Самостоятельная работа	Интерактивная лекция Занятия с использованием специализированного ПО Подготовка к занятию с использованием ЭБС
5.	Тема 5. Программирование классов на языке C++. Конструктор и деструктор класса. Наследование.	Лекция 5. Практическая работа 5 Самостоятельная работа	Интерактивная лекция Занятия с использованием специализированного ПО Подготовка к занятию с использованием ЭБС

5. Оценка планируемых результатов обучения

5.1 Система оценивания

Форма контроля	Макс. количество баллов	
	За одну работу	Всего
Текущий контроль:		
- защита практических работ	12 баллов	60 баллов
Промежуточная аттестация Экзамен		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

5.2 Критерии выставления оценки по дисциплине

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
100-83/ А, В	«отлично»/ «зачтено (отлично)»/ «зачтено»	<p>Выставляется обучающемуся, если он глубоко и прочно усвоил теоретический и практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся исчерпывающе и логически стройно излагает учебный материал, умеет увязывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения.</p> <p>Свободно ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «высокий».</p>
82-68/ С	«хорошо»/ «зачтено (хорошо)»/ «зачтено»	<p>Выставляется обучающемуся, если он знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей.</p> <p>Обучающийся правильно применяет теоретические положения при решении практических задач профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами.</p> <p>Достаточно хорошо ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной</p>

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
		<p>аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «хороший».</p>
67-50/ D, E	«удовлетворительно»/ «зачтено (удовлетворительно)»/ «зачтено»	<p>Выставляется обучающемуся, если он знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации. Обучающийся испытывает определённые затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами. Демонстрирует достаточный уровень знания учебной литературы по дисциплине. Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «достаточный».</p>
49-0/ F, FX	«неудовлетворительно»/ не зачтено	<p>Выставляется обучающемуся, если он не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации. Обучающийся испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами. Демонстрирует фрагментарные знания учебной литературы по дисциплине. Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации. Компетенции на уровне «достаточный», закреплённые за дисциплиной, не сформированы.</p>

Критерии оценивания / Уровень требований к обучающемуся	Макс. кол-во баллов
Текущий контроль, всего в т.ч.:	60
Практическая работа	5 -10
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
Промежуточная аттестация, всего в т.ч.:	40
Ответ на вопросы билета (2 вопроса – по 20 баллов максимум каждый)	20
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

5.3 Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

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\|a=%d b=%d c=%d x=%lf y=%lf
z=%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
\nsizeof(f)=%d\tsizeof(d)=%d\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\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.
2. Жизненный цикл программного обеспечения.
3. Обзор языков программирования.
4. Способы описания алгоритмов.
5. Структура программы на языке Си.
6. Типы данных в языке Си. Определение простых переменных в программе.
7. Арифметические операции языка Си.
8. Логические операции языка Си
9. Операторы языка Си: условный, оператор-переключатель.
10. Операторы циклов в языке Си.
11. Одномерные массивы в Си: определение, способы инициализации.
12. Типовые операции с одномерными массивами: поиск минимума, сумма элементов.
13. Двумерные массивы в языке Си: определение, способы инициализации.
14. Указатели: назначение, определение, инициализация.

15. Динамическое выделение памяти.
16. Функции ввода-вывода в C++.
17. Универсальный метод выделения динамической памяти под двумерные массивы.
18. Строки в языке Си: определение, инициализация, кодирование символов
19. Функции для работы со строками языка Си.
20. Структурный тип данных: определение типа и переменных, инициализация структур.
21. Вложенные структуры, массивы структур. Динамическое выделение памяти под массивы структур.
22. Перечислимый тип данных: определение, назначение, пример использования.
23. Динамические списки: назначение, основные принципы организации.
24. Односвязный список: определение, особенности использования, типовые алгоритмы обработки.
25. Сравнительный анализ односвязного списка и массива структур.
26. Двусвязный список: определение, особенности использования, типовые алгоритмы обработки.
27. Поточковый ввод-вывод. Определение потока, стандартные потоки, типы потоков.
28. Основные функции контроля состояния потоков.
29. Основные функции ввода строковой информации в поток и вывода информации из потока.
30. Форматированный потоковый ввод-вывод в языке Си.
31. Функции в Си: определение, прототип, вызов.
32. Основные преимущества структурного стиля программирования.
33. Формальные и фактические параметры. Связь формальных и фактических параметров функций.
34. Порядок вызова функции и возврата из нее. Передача параметров по значению и по адресу.
35. Функции и массивы: массивы как параметры функций.
36. Функции с умалчиваемыми значениями параметров.
37. Передача параметров функции main.
38. Ссылки: назначение, определение, использование.
39. Рекурсивные вызовы функций: назначение, примеры рекурсивных задач, прямая и косвенная рекурсия.
40. Разворачивание и свертывание рекурсии в памяти, глубина рекурсии.
41. Алгоритм быстрой обменной сортировки, его рекурсивная реализация
42. Указатели на функции: определение, назначение, примеры использования.
43. Указатели на функции как параметры функций.
44. Перегрузка функций: определение, достоинства, ограничения. Примеры реализации.
45. Препроцессор: директивы включения файлов и замены идентификаторов.
46. Препроцессор: директивы условной компиляции.
47. Принципы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм.
48. Определение классов и объектов в программах на языке Си++
49. Область видимости компонент класса.
50. Правила наследования классов в языке Си++.
51. Библиотеки `cstdlib`, `conio`, `iostream`, `math`, `algorithm`, `vector`, `cstdio`.
52. Операции ++ и --. Префиксное и постфиксное использование операций. Применение к указателям.

6. Учебно-методическое и информационное обеспечение дисциплины

6.1 Список источников и литературы

Литература

Основная

1. Дейл, Н. Программирование на C++ : учебник / Н. Дейл, Ч. Уимз, М. Хедингтон ; пер. с англ. А. С. Цемахмана. - 2-е изд. - Москва : ДМК Пресс, 2023. - 674 с. - (Учебник). - ISBN 978-5-89818-342-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2102629>
2. Корнеев, В. И. Программирование графики на C++. Теория и примеры : учебное пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. — Москва : ИНФРА-М, 2024. — 517 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Бакалавриат). — DOI 10.12737/23113. - ISBN 978-5-16-017914-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2111934>
3. Павловская, Т. А. C/C++. Структурное и объектно-ориентированное программирование : практикум / Т. А. Павловская, Ю. А. Щупак. - Санкт-Петербург : Питер, 2021. - 352 с. - (Серия «Учебное пособие»). - ISBN 978-5-4461-9799-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1857042>.

Дополнительная

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-prakticheskij-kurs-454165#page/2>
2. Хорев, П. Б. Объектно-ориентированное программирование с примерами на C# : учебное пособие / П.Б. Хорев. — Москва : ФОРУМ : ИНФРА-М, 2023. — 200 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-00091-680-3. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1926392>

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 Профессиональные базы данных и информационно-справочные системы

Доступ к профессиональным базам данных: <https://liber.rsuh.ru/ru/bases>

Информационные справочные системы:

1. Консультант Плюс
2. Гарант

7. Материально-техническое обеспечение дисциплины

Для обеспечения дисциплины используется материально-техническая база образовательного учреждения: учебные аудитории, оснащённые компьютером и проектором для демонстрации учебных материалов.

Состав программного обеспечения:

1. Windows
2. Microsoft Office
3. Kaspersky Endpoint Security

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

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- для слепых и слабовидящих: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением; письменные задания выполняются на компьютере со специализированным программным обеспечением или могут быть заменены устным ответом; обеспечивается индивидуальное равномерное освещение не менее 300 люкс; для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств; письменные задания оформляются увеличенным шрифтом; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

- для глухих и слабослышащих: лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования; письменные задания выполняются на компьютере в письменной форме; экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.

- для лиц с нарушениями опорно-двигательного аппарата: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением; письменные задания выполняются на компьютере со специализированным программным обеспечением; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены университетом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- для слепых и слабовидящих: в печатной форме увеличенным шрифтом, в форме электронного документа, в форме аудиофайла.

- для глухих и слабослышащих: в печатной форме, в форме электронного документа.

- для обучающихся с нарушениями опорно-двигательного аппарата: в печатной форме, в форме электронного документа, в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения:

- для слепых и слабовидящих: устройством для сканирования и чтения с камерой SARA CE; дисплеем Брайля PAC Mate 20; принтером Брайля EmBraille ViewPlus;
- для глухих и слабослышащих: автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих; акустический усилитель и колонки;
- для обучающихся с нарушениями опорно-двигательного аппарата: передвижными, регулируемые эргономическими партами СИ-1; компьютерной техникой со специальным программным обеспечением.

9. Методические материалы

9.1. Планы практических занятий

Тема 1. Основы программирования на языке С.

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 С/С++;
3. Подготовить развернутый отчет о проделанной работе.

Контрольные вопросы:

1. Структура программы на языке С.
2. Типы данных языка С. Приведение типов.
3. Арифметические и логические операции языка С.
4. Циклические структуры языка С: **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 С.

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 С: 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

Практическая 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 рекомендации 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	8	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	10	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	10	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	10	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	10	См. Список литературы, Интернет-ресурсов в разделе 6.
Промежуточная аттестация			
Подготовка к экзамену	См. Содержание дисциплины (раздел 3)	18	См. Примерные контрольные вопросы по курсу в разделе 5.4, Список литературы, Интернет- ресурсов в разделе 6
Итого по дисциплине		66	

9.3 Методические рекомендации по подготовке по подготовке письменных работ

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 С» 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 ДИСЦИПЛИНЫ

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 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.

Дисциплина направлена на формирование следующих компетенций:

ОПК-5 Способен использовать современные информационные технологии и программные средства при решении профессиональных задач; устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем.

ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения.

В результате освоения дисциплины обучающийся должен:

Знать: современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности; теоретические основы организации алгоритмов, структур данных и программ; операционные системы и оболочки; современные программные среды разработки информационных систем и технологий; основные языки программирования и современные программные среды разработки информационных систем и технологий.; методы разработки эффективных приложений и оптимизации по времени и памяти; главные управляющие структуры языков программирования; принципы программирования; алгоритмы отладки разрабатываемых приложений;

Уметь: выбирать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности; разрабатывать алгоритмы и программы для решения профессиональных задач; устанавливать программное и аппаратное обеспечение информационных и автоматизированных систем; применять принципы программирования для построения высокоэффективных приложений; проектировать и разрабатывать информационные прикладное программное обеспечение

Владеть: навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности; навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач; навыками установки программного и аппаратного обеспечения информационных и автоматизированных систем; методами оптимизации разрабатываемых алгоритмов по затрачиваемому машинному времени на исполнение программы и используемой памяти компьютера; основными управляющими структурами языков программирования; алгоритмами и структурами данных; методами и подходами к динамическому выделению памяти в

пространстве памяти компьютера методами создания программных прототипов для решения прикладных задач; базовыми принципами программирования на языке высокого уровня; навыками внедрения и адаптации программных ресурсов.

По дисциплине предусмотрена промежуточная аттестация – экзамен.

Общая трудоемкость освоения дисциплины составляет 4 зачетные единицы.