



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение
высшего образования

«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ФГБОУ ВО «ИГУ»

Кафедра метеорологии и физики околоземного космического пространства



Декан географического факультета
С.Ж. Воложжина
«10» июня 2019 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Наименование дисциплины **Б1.Б8 ПРОГРАММНЫЕ СРЕДСТВА АНАЛИЗА
СОСТОЯНИЯ ОКРУЖАЮЩЕЙ СРЕДЫ**

Направление подготовки 05.03.04 «Гидрометеорология»

Тип образовательной программы академический бакалавриат

Направленность (профиль) подготовки метеорология

Квалификация (степень) выпускника - бакалавр

Форма обучения очная/заочная

Согласовано с УМК географического
факультета

Протокол № 6 от «10» июня 2019 г.
Председатель С.Ж. Воложжина

Рекомендовано кафедрой метеорологии и
физики околоземного космического
пространства:

Протокол № 7 от «25» апреля 2019 г.
и.о.зав. кафедрой И.В. Латышева

Иркутск 2019 г.

Содержание

1. Цели и задачи дисциплины	3
2. Место дисциплины в структуре ООП	3
3. Требования к результатам освоения дисциплины	3
4. Объем дисциплины и виды учебной работы	3
5. Содержание дисциплины	4
5.1 Содержание разделов и тем дисциплины	4
5.2 Разделы дисциплины и междисциплинарные связи с обеспечиваемыми (последующими) дисциплинами	5
5.3 Разделы и темы дисциплины и виды занятий	5
5.4 Перечень лекционных занятий	6
6. Перечень семинарских, практических занятий, лабораторных работ, план самостоятельной работы студентов, методические указания по организации самостоятельной работы студентов	6
6.1 План самостоятельной работы студентов	7
6.2 Методические указания по организации самостоятельной работы студентов	8
7. Примерная тематика курсовых работ (проектов) (при наличии)	8
8. Учебно-методическое и информационное обеспечение дисциплины:	8
а) основная литература	8
б) дополнительная литература	8
в) программное обеспечение	8
г) базы данных, поисково-справочные и информационные системы	9
9. Материально-техническое обеспечение дисциплины	10
10. Образовательные технологии	10
11. Оценочные средства (ОС)	10

1. Цели и задачи дисциплины:

Цели: Формирование практических навыков по основам программирования на универсальном алгоритмическом языке высокого уровня C++ для решения профессиональных задач; овладение техникой постарения и реализации алгоритмов; развитие у студентов логического и аналитического мышления.

Задачи:

- обучить синтаксису и семантике, основным принципам и методам программирования на языке C++;
- научить использованию понятийного аппарата программирования;
- дать представление об основах структуризации задач в области гидрометеорологии;
- обучить практическим навыкам алгоритмизации и программирования с техническими и программными средствами для решения гидрометеорологических задач.

2. Место дисциплины в структуре ОПОП: Дисциплина «Программные средства анализа состояния окружающей среды» входит в базовую часть математического и естественнонаучного цикла, опирается на методы «Информатики» и «Высшей математики», её освоение необходимо для последующего изучения дисциплин «Геоинформатика», «Компьютерная графика» и «Автоматизированные методы обработки гидрометеорологической информации». Общая трудоёмкость составляет 2 зачётных единицы.

3. Требования к результатам освоения дисциплины (модуля):

Процесс изучения дисциплины (модуля) направлен на формирование следующих компетенций:
ОПК-1 владение базовыми знаниями в области фундаментальных разделов математики в объеме, необходимом для владения математическим аппаратом в гидрометеорологии, для обработки и анализа данных, прогнозирования гидрометеорологических характеристик
ОПК-6 способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности

В результате изучения дисциплины студент должен:

Знать: методы гидрометеорологических измерений, статистической обработки и анализа гидрометеорологических наблюдений с применением программных средств.

Уметь: решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.

Владеть: базовыми знаниями в области фундаментальных разделов математики в объеме, необходимом для владения математическим аппаратом в гидрометеорологии, для обработки и анализа данных, прогнозирования гидрометеорологических характеристик.

4. Объем дисциплины и виды учебной работы

Вид учебной работы	Всего часов	Семестры			
		3/2			
Аудиторные занятия (всего)	44/14	44/14			
в том числе:					
Лекции	14/4	14/4			
Практические занятия (ПЗ)	28/8	28/8			
Семинары (С)					
Лабораторные работы (ЛР)					
КСР	2/2	2/2			

Контроль	0/4	0/4			
Самостоятельная работа (всего)	28/54	28/54			
В том числе:					
Контрольные работы					
Курсовой проект (работа)					
Расчетно-графические работы	28/54	28/54			
Реферат (при наличии)					
<i>Другие виды самостоятельной работы</i>					
Вид промежуточной аттестации – зачет	-	-			
Контактная работа (всего)	44/14	44/14			
Общая трудоемкость	часы	72	72		
	зачётные единицы	2	2		

5. Содержание дисциплины

5.1. Содержание разделов и тем дисциплины. Все разделы и темы нумеруются

Раздел 1. Основы алгоритмизации и программирования

- 1.1. Основные этапы решения задач на ЭВМ. Понятие алгоритма. Методы формального описания алгоритмов. Схемы алгоритмов. Основные характеристики алгоритмов и этапы их разработки для решения простейших гидрометеорологических задач.
- 1.2. Базовые разновидности программных алгоритмов. Принципы алгоритмизации. Разветвленные и циклические алгоритмы. Сложные циклы. Алгоритмы с массивами. Взаимосвязь алгоритмов, моделей данных и постановок задач. Алгоритм и его программная реализация.
- 1.3. Понятие языка программирования. Основные парадигмы программирования – процедурное, логическое, функциональное. Синтаксис и семантика языка. Понятие алгоритмического языка программирования и наиболее распространенные представители универсальных алгоритмических языков высокого уровня. Информационные системы РОСГИДРОМЕТА, реализованные на программных продуктах Fortran и C++.
- 1.4. Функциональное содержание процессов компиляции (трансляции, интерпретации) и построения загрузочных модулей, отладочных операций и тестирования.

Раздел 2. Общая характеристика языка C++

- 2.1. История создания C++. Место языка C++ в общей иерархии алгоритмических языков программирования.
- 2.2. Реализация языка для различных вычислительных платформ и операционных сред. Интегрированная среда программирования системы Microsoft Visual C++.

Раздел 3. Основные элементы языка C++

- 3.1. Алфавит языка. Идентификаторы. Ключевые слова и символы. Знаки операций. Синтаксис описания констант и переменных. Основные типы данных.

Раздел 4. Структура программы на языке C++

- 4.1. Понятия программы, модуля, программной единицы. Общая структура программы. Пользовательские и библиотечные функции. Заголовочные файлы.
- 4.2. Препроцессор и его основные директивы. Комментарии.

Раздел 5. Операции и выражения

- 5.1. Арифметические операции. Операции инкрементации и декрементации. Логические операции и операции отношения. Операция условия. Операция присваивания.
- 5.2. Операция **sizeof**. Приоритет операций. Назначение выражений. Примеры выражений.

Раздел 6. Операторы управления

- 6.1. Основные виды операторов – операторы циклов, условных и безусловных переходов, оператор выбора.

6.2. Вспомогательные операторы. Простейшие операторы консольного ввода – вывода.

Раздел 7. Указатели, ссылки, массивы

7.1. Использование указателей как средства хранения адреса. Имена указателей. Операции над указателями. Оператор разыменования. Использование оператора адреса (&) при работе со ссылками. Возвращение значений с помощью ссылок.

7.2. Понятие массива. Синтаксис описания массивов. Обращение к элементам массива. Инициализация массивов. Массивы и указатели. Двумерные и одномерные массивы. Ввод и вывод массивов.

Раздел 8. Функции

8.1. Объявление и определение функций. Вызов функций. Формальные и фактические параметры. Механизм передачи параметров по значению и по адресу. Перегрузка функций. Глобальные и локальные переменные.

8.2. Область видимости и время жизни объектов. Классы памяти. Понятие рекурсии.

Раздел 9. Динамические данные

9.1. Модели памяти. Статические и динамические данные. Механизмы выделения, перераспределения и очистки динамической памяти. Функции, поддерживающие основные операции с динамической памятью.

9.2. Операторы **new** и **delete**. Динамические структуры данных. Линейные списки, стеки, очереди, бинарные деревья.

Раздел 10. Файлы и потоки

10.1. Описание и внутреннее представление файлов. Текстовые и бинарные файлы. Базовые операции над файлами. Режимы доступа. Позиционирование в файле.

10.2. Библиотечные функции работы с файлами. Понятие потока. Стандартные потоки в языке C++. Функции работы с потоками.

5.2 Разделы дисциплины и междисциплинарные связи с обеспечиваемыми (последующими) дисциплинами

№ п/п	Наименование обеспечиваемых (последующих) дисциплин	№№ разделов и тем данной дисциплины, необходимых для изучения обеспечиваемых (последующих) дисциплин (вписываются разработчиком)								
		1.3	4.1	7.2	9.1	10.1				
1.	Геоинформатика (Б1.Б.21)									
2.	Компьютерная графика (Б1.В.ДВ.2.2)	1.1	1.2	4.1	6.1	7.1	9.1	9.2	10.2	
3.	Автоматизированные методы обработки гидрометеорологической информации (Б1.В.ДВ.8.2)	1.1	1.2	4.1	6.1	7.1	9.1	9.2	10.1	10.2

5.3. Разделы и темы дисциплин (модулей) и виды занятий

№ п/п	Наименование раздела	Виды занятий в часах (очно/заочно)					
		Лекц.	Практ.	Семина.	Лаб.	СРС	Всего
1.	Раздел 1. Основы алгоритмизации и программирования	2	4				6
2.	Раздел 2. Общая характеристика языка C++	1/1	2/2				3/3

3.	Раздел 3. Основные элементы языка C++	1/1	2/2				3/3
4.	Раздел 4. Структура программы на языке C++	1	2			10/18	13/18
5.	Раздел 5. Операции и выражения	1/1	2/2				3/3
6.	Раздел 6. Операторы управления	1/1	2/2				3/3
7.	Раздел 7. Указатели, ссылки, массивы	2	4			4/8	10/8
8.	Раздел 8. Функции	1	2			4/8	7/8
9.	Раздел 9. Динамические данные	2	4			5/10	11/10
10.	Раздел 10. Файлы и потоки	2	4			5/10	11/10
	Итого:	14/4	28/8			28/54	70/66
	КСР						2/2
	Контроль						0/4
	ВСЕГО						72/72

5.4 Перечень лекционных занятий

№ п/п	№ раздела и темы дисциплины (модуля)	Наименование используемых технологий	Трудоемкость (часы) (очно/заочно)	Оценочные средства	Формируемые компетенции
1	2	3	4	5	6
1.	1	Основы алгоритмизации и программирования	2	Оценка в баллах	ОПК-1 ОПК-6
2.	2	Общая характеристика языка C++	1/1	Оценка в баллах	ОПК-1 ОПК-6
3.	3	Основные элементы языка C++	1/1	Оценка в баллах	ОПК-1 ОПК-6
4.	4	Структура программы на языке C++	1	Оценка в баллах	ОПК-1 ОПК-6
5.	5	Операции и выражения	1/1	Оценка в баллах	ОПК-1 ОПК-6
6.	6	Операторы управления	1/1	Оценка в баллах	ОПК-1 ОПК-6
7.	7	Указатели, ссылки, массивы	2	Оценка в баллах	ОПК-1 ОПК-6
8.	8	Функции	1	Оценка в баллах	ОПК-1 ОПК-6
9.	9	Динамические данные	2	Оценка в баллах	ОПК-1 ОПК-6
10.	10	Файлы и потоки	2	Оценка в баллах	ОПК-1 ОПК-6
	Итого:		14/4 ч		

6. Перечень семинарских, практических занятий и лабораторных работ

№ п/п	№ раздела и (темы дисциплины)	Наименование семинаров, практических и лабораторных работ	Трудоемкость (час.)	Оценочные средства	Формируемые компетенции

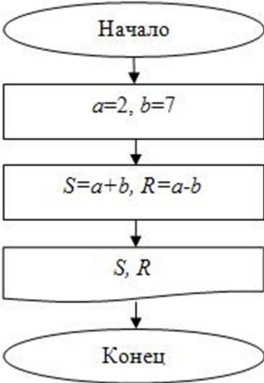
1	2	3	4	5	6
1.	1	<p>Составление алгоритма генерирования таблицы умножения</p> <p><i>С помощью оператора цикла for, разработать программу, которая будет выводить таблицу умножения введенного пользователем числа с клавиатуры.</i></p> <pre> #include <iostream> using namespace std; int main() { setlocale (LC_ALL, "RUS"); int chislo; cout<<"\t\t\tТаблица умножения введенного числа\n"; cout<<"\t\t\t----- -----\n"; cout<<endl; cout<<"Введите число: "; cin>>chislo; cout<<endl; for (int j=1; j<=10; j++) { cout<<j<<" * "<<chislo<<" = "<<j*chislo<<endl; } cout<<endl; return 0; } </pre>	4	Оценка в баллах	ОПК-1 ОПК-6
2.	2	<p>Обзор системы Microsoft Visual C++</p> <p><i>Интегрированная среда разработки Visual Studio — это оригинальная среда запуска, которая позволяет редактировать, отлаживать и создавать код, а затем публиковать приложения. Интегрированная среда разработки (IDE) — это многофункциональная программа, которую можно использовать для различных аспектов разработки программного обеспечения. Помимо стандартного редактора и отладчика, которые существуют в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства выполнения кода, графические конструкторы и многие другие функции для упрощения процесса разработки программного обеспечения.</i></p>	2/2	Оценка в баллах	ОПК-1 ОПК-6
3.	3	<p>Работа с основными типами данных C++</p>	2/2	Оценка в баллах	ОПК-1 ОПК-6

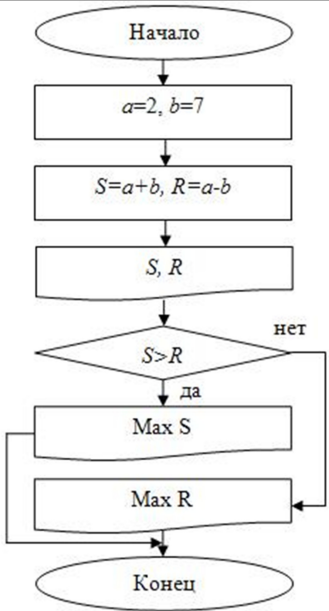
		Типы данных	Размер (байт)	Диапазон значений		
		bool	1	false, true		
		signed char	1	-128 ... +127		
		unsigned char	1	0 ... +255		
		signed short int	2	-32 768 ... +32 767		
		unsigned short int	2	0 ... 65 535		
		signed int	4	-2 147 483 648 ... 2 147 483 647		
		unsigned int	4	0 ... 4 294 967 295		
		signed long int	8	-9 223 372 036 854 775 808 ... 9 223 372 036 854 775 807		
		unsigned long int	8	2 ⁶⁴		
		float	4	-2 147 483 648.0 ... 2 147 483 647.0		
		double	8	-9 223 372 036 854 775 808.0 ... 9 223 372 036 854 775 807.0		
		long double	16	2.2E-308...1.8E+308		
<small>(типы данных представлены на основе среды разработки Xcode)</small>						
4.	4	<p>Обзор заголовочных файлов и библитек</p> <p><i>В языке программирования C++ термин Стандартная Библиотека означает коллекцию классов и функций, написанных на базовом языке. Стандартная Библиотека поддерживает несколько основных контейнеров, функций для работы с этими контейнерами, объектов-функции, основных типов строк и потоков (включая интерактивный и файловый ввод-вывод), поддержку некоторых языковых особенностей, и часто используемые функции для выполнения таких задач, как, например, нахождение квадратного корня числа.</i></p> <p>Ввод-вывод:</p> <ul style="list-style-type: none"> • <i>Fstream</i> • <i>iomanip</i> • <i>ios</i> • <i>iosfwd</i> • <i>iostream (ostream, istream)</i> • <i>sstream</i> • <i>streambuf</i> 		2	Оценка в баллах	ОПК-1 ОПК-6
5.	5	<p>Обзор работы операций и примеры выражений C++</p> <p><i>Здесь мы разберем predefined операции, такие, как сложение, вычитание, сравнение и т.п., рассмотрим их приоритеты. Скажем, результатом выражения 3+4*5 является 23, а не 35 потому, что операция умножения (*) имеет более высокий приоритет, чем операция сложения (+). Кроме того, мы обсудим вопросы преобразований типов данных – и явных, и неявных. Например, в выражении 3+0.7 целое значение 3 станет вещественным перед выполнением операции сложения. Выражение состоит из одного или более операндов, в простейшем случае – из одного литерала или объекта. Результатом такого выражения является r-значение его операнда. Например:</i></p>		2/2	Оценка в баллах	ОПК-1 ОПК-6

		<pre>void mumble() { 3.14159; "melancholia"; upperBound; } </pre> <p>Результатом вычисления выражения 3.14159 станет 3.14159 типа double, выражения "melancholia" – адрес первого элемента строки типа const char*. Значение выражения upperBound – это значение объекта upperBound, а его типом будет тип самого объекта. Более общим случаем выражения является один или более операндов и некоторая операция, применяемая к ним:</p> <pre>salary + raise ivec[size/2] * delta first name + " " + last name </pre>		
6.	6	<p>Операторы ввода/вывода В C++, как и в C, нет встроенных в язык средств ввода-вывода. В C для этих целей используется библиотека <code>stdio.h</code>. В C++ разработана новая библиотека ввода-вывода <code>iostream</code>, использующая концепцию объектно-ориентированного программирования:</p> <pre>#include <iostream> </pre> <p>Библиотека <code>iostream</code> определяет три стандартных потока: <code>cin</code> стандартный входной поток (<code>stdin</code> в C) <code>cout</code> стандартный выходной поток (<code>stdout</code> в C) <code>cerr</code> стандартный поток вывода сообщений об ошибках (<code>stderr</code> в C) Для их использования в <i>Microsoft Visual Studio</i> необходимо прописать строку: <code>using namespace std;</code> Для выполнения операций ввода-вывода переопределены две операции поразрядного сдвига: <code>>></code> получить из входного потока <code><<</code> поместить в выходной поток</p>	2/2	Оценка в баллах ОПК-1 ОПК-6
7.	7	<p>Инициализация и заполнение массивов различных типов Пример инициализации массива</p> <pre>string students[10] = { "Иванов", "Петров", "Сидоров", "Ахмедов", "Ерошкин", "Выхин", "Андреев", "Вин Дизель", "Картошкин", "Чубайс" }; </pre> <p>Описание синтаксиса Массив создается почти так же, как и обычная переменная. Для хранения десяти фамилий нам нужен массив, состоящий из 10 элементов. Количество элементов массива задается при его объявлении и заключается в квадратные скобки. Чтобы описать элементы массива сразу при его создании, можно использовать фигурные скобки. В</p>	4	Оценка в баллах ОПК-1 ОПК-6

		<i>фигурных скобках значения элементов массива перечисляются через запятую. В конце закрывающей фигурной скобки ставится точка с запятой.</i>			
8.	8	Составление функций для работы с таблицами ТМС <i>Необходимо воспользоваться полученными знаниями при работе с функциями стандартной библиотеки, операторов ввода-вывода и работы с массивами, чтобы написать функции, реализующие алгоритм работы с таблицами метеорологических данных ТМС-84</i>	2	Оценка в баллах	ОПК-1 ОПК-6
9.	9	Инициализация динамического массива для работы с ТМС <i>Необходимо воспользоваться полученными знаниями при работе с функциями стандартной библиотеки, операторов ввода-вывода и работы с массивами (динамическими), чтобы написать функции, реализующие алгоритм работы с таблицами метеорологических данных ТМС-84</i>	4	Оценка в баллах	ОПК-1 ОПК-6
10.	10	Создание файлов электронных таблиц на основе ТМС <i>С помощью языка C++ можно работать со сторонними программами. Требуется только библиотека-парсер, в которой будут описаны способы взаимодействия с этой программой.</i> <pre>{ string word; Variant exl; exl = CreateOleObject("Excel.Application") ; exl.OlePropertyGet("Workbooks"). OleProcedure("Open", "D:\\1.xls"); for(int i = 1 ; i < 5; i++) for(int j = 1; j < 4; j++) word=exl.OlePropertyGet("Cells", i, j); }</pre> <i>Программа выполняет следующую задачу: посредством Ole создает объект (CreateOleObject()), распознаваемый как объект документа Excel. После этого объект exl открывает файл по заданному пути и интерпретирует его (OleProcedure("Open", "D:\\1.xls"). Затем в переменную word выдаются значения клеток. Итак, при помощи команд OlePropertyGet, OlePropertySet и OleProcedure можно получить доступ ко всем объектам и коллекциям Excel.</i>	4	Оценка в баллах	ОПК-1 ОПК-6
	Итого:		28/8 ч		

6.1. План самостоятельной работы студентов

№ нед.	Тема	Вид самостоятельной работы и проверка задания	Рекомендуемая литература	Кол-во часов (очно/заочно)
1-3	4	<p>Программная реализация линейных алгоритмов. (форма отчётности – расчетная работа)</p> <p>Пример 1. Даны числа $a = 2, b = 7$. Вычислить сумму S и разность R чисел a и b.</p> <p>Приведем блок-схему на рис. 1.</p>  <pre> graph TD Start([Начало]) --> Init[a=2, b=7] Init --> Calc[S=a+b, R=a-b] Calc --> Output[S, R] Output --> End([Конец]) </pre> <p>Рис. 1. Блок-схема для примера 1</p> <p>Код программы (Visual Studio):</p> <pre> // progal0.cpp: определяет точку входа для консольного приложения. // #include "stdafx.h" #include <iostream> using namespace std; int main() { double a, b, S, R; a=2; b=7; S=a+b; R=a-b; cout<<"S="<<S<<endl; cout<<"R="<<R<<endl; return 0; } </pre>	1-о, 2-о, 3-о, 1-д, 2-д, 3-д.	5/9
4-6	4	<p>Программная реализация разветвленных алгоритмов. (форма отчётности – расчетная работа)</p> <p>Пример 2. Даны числа $a = 2, b = 7$. Вычислить сумму S и разность R чисел a и b. Сравнить полученные значения S и R и указать большее из них.</p> <p>Блок-схема представлена на рис. 2.</p>	1-о, 2-о, 3-о, 1-д, 2-д, 3-д.	5/9

		 <pre> graph TD Start([Начало]) --> Init[a=2, b=7] Init --> Calc[S=a+b, R=a-b] Calc --> Print[S, R] Print --> Dec{S>R} Dec -- да --> MaxS[Max S] Dec -- нет --> MaxR[Max R] MaxS --> End([Конец]) MaxR --> End </pre> <p>Рис. 2. Блок-схема для примера 2</p> <p>Код программы (Visual Studio):</p> <pre> // progal4.cpp: определяет точку входа для консольного приложения. // #include "stdafx.h" #include <iostream> using namespace std; int main() { double a, b, S, R; a=2; b=7; S=a+b; R=a-b; cout<<"S="<<S<<" R="<<R<<endl; if (S>R) cout<<"Max S"<<endl; else cout<<"Max R"<<endl; return 0; } </pre>		
7-9	7	<p>Программная реализация алгоритмов с массивами. (форма отчётности – расчетная работа) <i>Ввести целочисленный массив из N элементов с клавиатуры. Отсортировать его по возрастанию методом пузырька.</i></p> <p><i>Исходный код на языке C++</i> <i>/*</i> <i>* Ввести целочисленный массив из N целых чисел.</i> <i>* Отсортировать этот массив по возрастанию методом пузырька</i> <i>*/</i> #include <iostream> using namespace std; int main() { int *arr; // указатель для выделения памяти под массив int size; // размер массива // Ввод количества элементов массива </p>	1-о, 2-о, 3-о, 1-д, 2-д, 3-д.	4/8

		<pre> cout << "n = "; cin >> size; if (size <= 0) { // Размер массива должен быть // положительным cerr << "Invalid size" << endl; return 1; } arr = new int[size]; // выделение памяти // под массив // заполнение массива for (int i = 0; i < size; i++) { cout << "arr[" << i << "] = "; cin >> arr[i]; } int temp; // временная переменная для // обмена элементов местами // Сортировка массива пузырьком for (int i = 0; i < size - 1; i++) { for (int j = 0; j < size - i - 1; j++) { if (arr[j] > arr[j + 1]) { // меняем элементы местами temp = arr[j]; arr[j] = arr[j + 1]; arr[j + 1] = temp; } } } // Вывод отсортированного массива на // экран for (int i = 0; i < size; i++) { cout << arr[i] << " "; } cout << endl; delete [] arr; // освобождение памяти; return 0; } </pre>		
10-11	8	<p>Программная реализация алгоритмов с использованием функций и указателей. (форма отчёта – расчетная работа)</p> <p><i>Функция <code>scanf</code> предназначена для ввода данных с клавиатуры.</i></p> <p><i>Для применения данной функции следует сначала подключить библиотеку языка C для операций ввода и вывода. Выглядит это следующим образом:</i></p> <pre>#include <stdio.h></pre> <p><i>Эта же библиотека требуется для функции <code>printf</code>.</i></p> <p><i>Программу для ввода и вывода массива можно представить таким образом:</i></p> <pre>#include <stdio.h> #include <conio.h> // нужно для подключения getch main() { int i, z, m[10]; // Объявляем переменные // целого типа и массив из 10 ячеек for(i = 0; i < 10; i++) // Начинаем цикл // для заполнения массива { scanf("%d", &z); // Считываем с // клавиатуры значение и присваиваем его // переменной z m[i] = z; // Присваиваем ячейке // массива введенное значение } } </pre>	1-о, 2-о, 3-о, 1-д, 2-д, 3-д.	4/8

		<pre> } for(i = 0; i < 10; i++) // Запускаем цикл для вывода массива на экран printf("m[%d] = %d\n", i, m[i]); // Функция вывода getch(); //Программа не закроется пока мы на нажмем любую кнопку } </pre>		
12-13	9	<p>Программная реализация динамических типов данных. (форма отчётности – расчетная работа)</p> <p>При объявлении статического массива, его размером должна являться числовая константа, а не переменная. В большинстве случаев целесообразно выделять определенное количество памяти для массива, значение которого изначально неизвестно.</p> <p>Например, необходимо создать динамический массив из N элементов, где значение N задается пользователем. В предыдущем уроке мы учились выделять память для переменных, используя указатели. Выделение памяти для динамического массива имеет аналогичный принцип.</p> <p>Создание динамического массива</p> <pre> #include <iostream> using namespace std; int main() { int num; // размер массива cout << "Enter integer value: "; cin >> num; // получение от пользователя размера массива int *p_darr = new int[num]; // Выделение памяти для массива for (int i = 0; i < num; i++) { // Заполнение массива и вывод значений его элементов p_darr[i] = i; cout << "Value of " << i << " element is " << p_darr[i] << endl; } delete [] p_darr; // очистка памяти return 0; } </pre>	1-о, 2-о, 3-о, 1-д, 2-д, 3-д.	5/10
14-16	10	<p>Программная реализация операций файлового ввода – вывода. (форма отчётности – расчетная работа)</p> <p><i>Пример: необходимо создать текстовый файл и записать в него строку Работа с файлами в C++. Для этого необходимо проделать следующие шаги:</i></p> <ul style="list-style-type: none"> • <i>создать объект класса ofstream;</i> • <i>связать объект класса с файлом, в который будет производиться запись;</i> • <i>записать строку в файл;</i> • <i>закрывать файл.</i> <p><i>Почему необходимо создавать объект класса ofstream, а не класса ifstream? Потому, что нужно сделать запись в файл, а если бы нужно было считать данные из файла, то создавался бы объект класса ifstream.</i></p> <pre> // создаём объект для записи в файл ofstream /*имя объекта*/; // объект класса ofstream </pre> <p><i>Назовём объект – fout, Вот что получится:</i></p>	1-о, 2-о, 3-о, 1-д, 2-д, 3-д.	5/10

	<pre>ofstream fout;</pre> <p>Для чего нам объект? Объект необходим, чтобы можно было выполнять запись в файл. Уже объект создан, но не связан с файлом, в который нужно записать строку.</p> <pre>fout.open("cppstudio.txt"); //</pre> <p>связываем объект с файлом</p> <p>Через операцию точка получаем доступ к методу класса open(), в круглых скобках которого указываем имя файла. Указанный файл будет создан в текущей директории с программой. Если файл с таким именем существует, то существующий файл будет заменен новым. Итак, файл открыт, осталось записать в него нужную строку. Делается это так:</p> <pre>fout << "Работа с файлами в C++";</pre> <p>// запись строки в файл</p> <p>Используя операцию передачи в поток совместно с объектом fout строка Работа с файлами в C++ записывается в файл. Так как больше нет необходимости изменять содержимое файла, его нужно закрыть, то есть отделить объект от файла.</p> <pre>fout.close(); // закрываем файл</pre> <p>Итог – создан файл со строкой Работа с файлами в C++.</p>		
Итого:			28/54 ч

Пояснения. В указанной литературе: о – основная, д – дополнительная.

Примечание:

- а) Темы для самостоятельной работы могут быть предложены самими студентами.
- б) Для итоговой аттестации по курсу необходимо выполнить один аналитический обзор и один реферат или расчётную работу. *Для самостоятельной работы рекомендуется использовать также*

Реферативные журналы: Сводные тома журналов География. Геофизика; Охрана окружающей среды и воспроизводство природных ресурсов;

Периодические научные статьи в журналах: География и природные ресурсы, Оптика атмосферы и океана, География, Метеорология и гидрология, Известия Иркутского государственного университета (серия Науки о Земле) и др.

6.2. Методические указания по организации самостоятельной работы студентов

Для выполнения всех перечисленных самостоятельных работ студенту предоставляется возможность использования: одного из трех компьютерных классов во внеучебное время (предварительная запись у дежурных в классе, все компьютеры подключены к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду университета), фондов стационарной библиотеки в 6-м корпусе и фундаментальной библиотеки ИГУ, читальных залов Института академии наук (согласно заключенным с ними Договорами), фондов библиотеки Иркутского управления по гидрометеорологии и мониторингу окружающей среды, индивидуальных консультаций с преподавателями факультета (согласно графику еженедельных консультаций).

- 7. **Примерная тематика курсовых работ (проектов) (при наличии)** не предусмотрена учебным планом

8. Учебно-методическое и информационное обеспечение дисциплины:

а) основная литература

1. Огнева, М. В. Программирование на языке С++: практический курс : учебное пособие для бакалавриата и специалитета / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2019. — 335 с. — (Бакалавр и специалист). — ISBN 978-5-534-05123-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://www.biblio-online.ru/bcode/438987> (дата обращения: 22.10.2019).
2. Зыков, С. В. Программирование. Функциональный подход : учебник и практикум для академического бакалавриата / С. В. Зыков. — Москва : Издательство Юрайт, 2019. — 164 с. — (Бакалавр. Академический курс). — ISBN 978-5-534-00844-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://www.biblio-online.ru/bcode/434613> (дата обращения: 22.10.2019).
3. Зыков, С. В. Программирование. Объектно-ориентированный подход : учебник и практикум для академического бакалавриата / С. В. Зыков. — Москва : Издательство Юрайт, 2019. — 155 с. — (Бакалавр. Академический курс). — ISBN 978-5-534-00850-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://www.biblio-online.ru/bcode/434106> (дата обращения: 22.10.2019).

б) дополнительная литература

1. Трофимов, В. В. Алгоритмизация и программирование : учебник для академического бакалавриата / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2019. — 137 с. — (Бакалавр. Академический курс. Модуль). — ISBN 978-5-534-07834-3. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://www.biblio-online.ru/bcode/423824> (дата обращения: 22.10.2019).
2. Кувшинов, Д. Р. Основы программирования : учебное пособие для вузов / Д. Р. Кувшинов. — Москва : Издательство Юрайт, 2019 ; Екатеринбург : Изд-во Урал. ун-та. — 104 с. — (Университеты России). — ISBN 978-5-534-07559-5 (Издательство Юрайт). — ISBN 978-5-7996-1411-9 (Изд-во Урал. ун-та). — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://www.biblio-online.ru/bcode/441475> (дата обращения: 22.10.2019).
3. Лебедев, В. М. Программирование на vba в ms excel : учебное пособие для академического бакалавриата / В. М. Лебедев. — Москва : Издательство Юрайт, 2019. — 272 с. — (Высшее образование). — ISBN 978-5-9916-7880-3. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://www.biblio-online.ru/bcode/433415> (дата обращения: 22.10.2019).

в) программное обеспечение

1. Программное обеспечение Microsoft Visual Studio (Express Edition)
2. Kaspersky Endpoint Security для бизнеса - Стандартный Russian Edition – Лицензия № 1B08161103014721370444 от 03.11.2016 г. – 27 экз.

Во время проведения лабораторных работ студенты могут использовать возможности компьютерного класса с выходом в интернет. Все задания для лабораторных работ представлены на Образовательном портале ИГУ (сайт educa.isu.ru). Студенты, которые не смогли присутствовать на занятии по уважительной причине, могут выполнить работу удаленно.

г) базы данных, информационно-справочные и поисковые системы

1. Образовательном портале ИГУ - <https://educa.isu.ru>
2. Архив метеорологических данных NCEP/NCAR Reanalysis - <https://www.esrl.noaa.gov>
3. Справка и обучение Microsoft Office <https://support.office.com/ru-ru/>
4. Каждый студент обеспечен индивидуальным доступом к информационно-телекоммуникационной сети «Интернет» к следующим электронно-библиотечным системам:
 - (электронным библиотекам – ЭБС; электронный читальный зал - ЭЧЗ);
 - ЭБС «Издательство Лань»;
 - ЭБС ЭЧЗ «Библиотех»;
 - ЭБС «Национальный цифровой ресурс «Руконт»;
 - ЭБС «Айбукс.ru/ibooks.ru»;
 - Электронная библиотека «Интуит.ру»;
 - Электронная библиотека «Академия»;
 - Электронно-библиотечная система «ЭБС Юрайт»;
 - Электронная библиотека диссертаций РГБ;
 - ЭБС «Научная электронная библиотека «ELIBRARY.RU»;
 - ЭКБСОН Государственная информационная система «Национальная электронная библиотека» (НЭБ).

9. Материально-техническое обеспечение дисциплины (модуля):

1. Учебная аудитория с мультимедийным проектором для проведения лекционных занятий для потока.
2. Учебная аудитория с мультимедийным проектором для проведения семинарских занятий по учебным группам.
3. Компьютерный класс с доступом в интернет.

10. Образовательные технологии:

Соответствующий тематике занятия иллюстрационный материал, переведенный в электронный формат и оформленный в виде презентаций. Для демонстрации данных презентаций привлекается мультимедиа оборудование.

В своей самостоятельной работе студенты могут использовать возможности компьютерного класса с выходом в интернет.

11. Оценочные средства (ОС):

- 11.1. Оценочные средства для входного контроля – не предусмотрены
- 11.2. Оценочные средства текущего контроля – тесты, контрольные работы, контроль выполнения практических работ

Программа оценивания контролируемой компетенции:

Раздел/ Тема	Индекс и уровень формируемой компетенции	ОС	Содержание задания
Раздел 1. Основы алгоритмизации и программирования	ОПК-1 ОПК-6	С, УО, П, К	Составить и обсудить на занятии проблемные вопросы по изученному разделу, заслушать презентации и устные доклады, обсудить решение практических заданий.
Раздел 2. Общая характеристика языка С++	ОПК-1 ОПК-6	С, УО, П, К	Составить и обсудить на занятии проблемные вопросы по изученному разделу, заслушать презентации и устные доклады, обсудить решение практических заданий.
Раздел 3. Основные элементы языка С++	ОПК-1 ОПК-6	С, УО, П, К	Составить и обсудить на занятии проблемные вопросы по изученному разделу, заслушать презентации и устные доклады, обсудить решение практических заданий.
Раздел 4. Структура программы на языке С++	ОПК-1 ОПК-6	С, УО, П, К	Составить и обсудить на занятии проблемные вопросы по изученному разделу, заслушать презентации и устные доклады, обсудить решение практических заданий.
Раздел 5. Операции и выражения	ОПК-1 ОПК-6	С, УО, П, К	Составить и обсудить на занятии проблемные вопросы по изученному разделу, заслушать презентации и устные доклады, обсудить решение практических заданий.
Раздел 6. Операторы управления	ОПК-1 ОПК-6	С, УО, П, К	Составить и обсудить на занятии проблемные вопросы по изученному разделу, заслушать презентации и устные доклады, обсудить решение практических заданий.
Раздел 7. Указатели, ссылки, массивы	ОПК-1 ОПК-6	С, УО, П, К	Составить и обсудить на занятии проблемные вопросы по изученному разделу, заслушать презентации и устные доклады, обсудить решение практических заданий.
Раздел 8. Функции	ОПК-1 ОПК-6	С, УО, П, К	Составить и обсудить на занятии проблемные вопросы по изученному разделу, заслушать презентации и устные доклады, обсудить решение практических

			заданий.
Раздел 9. Динамические данные	ОПК-1 ОПК-6	С, УО, П, К	Составить и обсудить на занятии проблемные вопросы по изученному разделу, заслушать презентации и устные доклады, обсудить решение практических заданий.
Раздел 10. Файлы и потоки	ОПК-1 ОПК-6	С, УО, П, К	Составить и обсудить на занятии проблемные вопросы по изученному разделу, заслушать презентации и устные доклады, обсудить решение практических заданий.

Примечание:

УО – устный опрос (собеседование)

П – практическая работа

К – контрольная работа

С – собеседование, включая по презентации.

11.3. Оценочные средства для промежуточной аттестации (в форме экзамена или зачета) – в соответствии с учебным планом очного отделения: 3 семестр – зачет.

Материалы для проведения текущего и промежуточного контроля знаний студентов:

• **Демонстрационные варианты:**

Демонстрационный вариант контрольной работы №1

1. Составить блок- схему алгоритма и программу вычисления длины окружности и площади круга.
2. Составить блок-схему алгоритма и программу вычисления функции $Y = EXP(-2X)$ при $X > 0$ и $Y = EXP(2X)$ при $X < 0$.
3. Составить блок- схему алгоритма и программу вычисления вещественных корней квадратного уравнения $AX^2 + BX + C = 0$.
4. Составить блок- схему алгоритма и программу вычисления функции $Y = SQRT(X^2 + 3) + LOG10(ABS(X - 2.3))$ при $X \geq 3$ и $Y = ((X^2 + 3)^2 + LOG10(ABS(X - 2.3)))$ при $X < 3$.
5. Составить блок-схему алгоритма и программу вычисления функции $Y = X^2 / (A + B)$ при изменении X от 1 до 10 с шагом 2 .

Демонстрационный вариант теста №1

```
1)
int i = 10;
for(int b = 0; b < 9; ++b)
    i++;
cout << i << endl;
```

Что выведется на экран?

- a) 10
- b) 18
- c) 19
- d) 20

2)

```
int b = 12;
int a = 7;
for(int i = 0; i < 4; ++i)
    b += 2;
    a++;
cout << a << " " << b << endl;
```

Что выведется на экран?

- a) 7 12
- b) 8 20
- c) 11 20
- d) 12 20

3)

```
int b = 10;
int a = 5;
int i = 0;
while(i < 6)
{
    ++i;
    ++a;
    b += 3;
}
cout << a << " " << b << " " << i << endl;
```

Что выведется на экран?

- a) 11 28 6
- b) 12 30 5
- c) 11 28 5
- d) 11 28 0

4)

```
int b = 10;
int a = 5;
int i = 0;
do
{
    ++i;
    ++a;
    b += 3;
} while(i < 6);
cout << a << " " << b << " " << i << endl;
```

Что выведется на экран?

- a) 11 28 6
- b) 12 30 5
- c) 11 28 5
- d) 11 28 0

5)

```
int b = 10;
int a = 5;
```

```

int i = 0;
do
{
    ++i;
    a += 1;
    b += 4;
} while(i < 6);
cout << a << " " << b << " " << i << endl;

```

Что выведется на экран?

- a) 11 33 7
- b) 12 30 5
- c) 10 33 5
- d) 11 33 6

Вопросы для собеседования и коллоквиумов приведены в пункте 6.1.

Требования к зачету

По каждой дисциплине направления преподаватель разрабатывает собственную шкалу оценок. Обучающийся получает зачет по дисциплине, если в течение семестра он набирает не менее 60 баллов. По указанной дисциплине применяется следующая шкала

Баллы, полученные обучающимися по дисциплине в течение семестра	Академическая оценка	
	60-70 баллов	удовлетворительно
71-85 баллов	хорошо	
86-100 баллов	отлично	

Если количество баллов, которое наберет обучающийся в течение семестра, будет недостаточным для получения им положительного результата, преподаватель вправе потребовать от обучающегося выполнения дополнительных заданий. Решение о возможности и форме выполнения обучающимся дополнительных заданий для получения большего количества баллов принимается преподавателем.

При активном участии обучающегося в научной работе (тезисы, статьи, выступления на конференциях) преподаватель может добавить до 10 бонусных баллов, но общее количество баллов не должно превышать 100.

Если студент выполнил все задания удовлетворительно и не имеет пропусков занятий, то зачет ставится автоматически.

Если студент имеет какие-либо «долги», то он может их погасить в отведенное преподавателю время на индивидуальную работу со студентами (расписание консультаций – на доске объявлений, в деканате и на кафедре).

Примерный перечень вопросов к зачету

1. Перечислите основные этапы процесса решения задачи на ЭВМ. Приведите примеры прямых, разветвленных и циклических алгоритмов. Укажите область использования циклических алгоритмов с неизвестным числом повторений. Приведите примеры циклических алгоритмов со сложными циклами.
2. Дайте определение массива. Как определить адрес элемента в двумерном массиве?
3. В чем разница между компилятором и интерпретатором? Как происходит компиляция исходного кода программы? В чем состоит назначение компоновщика?
4. Почему C++ стал стандартом в области разработки программных продуктов? Перечислите основные возможности интегрированной среды программирования Microsoft Visual C++. Дайте характеристику основным компонентам среды программирования. Перечислите элементы главного меню и опишите их назначение.

5. Перечислите основные составляющие, входящие в структуру программы на языке C++. Наличие какой функции является обязательным условием в любой программе на языке C++?
6. В чем состоит назначение препроцессора? Что такое функция и каково ее назначение? Какую роль в программе выполняют библиотечные функции? Перечислить типы комментариев, и чем они отличаются друг от друга?
7. Какие наборы знаков образуют алфавит языка? Какие правила необходимо выполнять при написании идентификаторов? Дайте определение константы и переменной.
8. Перечислите основные типы данных. В чем разница между целочисленными и вещественными переменными? В чем особенности символьной переменной? Почему необходимо объявлять переменные?
9. Какой операции соответствует символ %? В чем разница между операциями инкремента и декремента? Укажите на отличие префиксной и постфиксной форм записи операций.
10. Перечислите основные операции отношения. Что является результатом любой операции отношения? Приведите пример практической реализации операции условия.
11. Перечислите особенности операции присваивания в языке C++. Что такое выражение?
12. Операторы циклов, используемых в языке C++. В чем разница между операторами циклов `do while` и `while`? Запишите общую структуру оператора цикла `for`. Приведите пример использования оператора `for` для “убывающего” цикла. Как осуществить принудительный выход из цикла, организованного оператором `for` ?
13. Структура оператора `switch` . Запишите возможные структуры оператора условного перехода `if`. Приведите конкретные примеры таких структур.
14. Особенности использования операторов ввода – вывода `cin` и `cout` в языке C++?
15. Оператор для получения адреса переменной? Какой оператор позволяет получить значение, записанное по адресу, содержащемуся в указателе? В чем разница между адресом, который хранится в указателе, и значением, записанным по этому адресу? В чем различие между оператором разыменования и оператором получения адреса?
16. Покажите разницу между ссылкой и указателем. Как обратиться к первому и последнему элементу массива? Операторы для определения свойств массивов (размер, кол-во элементов).
17. В каких случаях целесообразно использовать ключевое слово `typedef` для переименования типов? Покажите структуру описания переменной перечисляемого типа. Приведите примеры использования переменных перечисляемого типа.
18. Чем отличается структура от массива? Как провести инициализацию элементов структуры? Как осуществляется доступ к элементам структуры? Чем отличаются объединения от структур?
19. В чем разница между объявлением прототипа и определением функции? Должны ли имена параметров, указанные в определении и вызове функции, соответствовать друг другу? Как следует объявлять функцию, если она не возвращает значения?
20. Что такое локальная переменная, область видимости, перегрузка функции, рекурсия?
21. Когда следует использовать глобальные переменные? К какому классу памяти будет относиться переменная по умолчанию?
22. Что такое динамические данные? Каким образом выделяется память под динамические данные? Перечислите основные особенности динамической памяти. Что возвращает оператор `new` после выделения участка динамической памяти? Каково назначение оператора `delete`? Перечислите основные операции с динамическими структурами данных.
23. В чем особенности однонаправленных и двунаправленных линейных списков? Как осуществляется выборка элементов данных в стеке? Опишите организацию движения данных в очереди. Укажите преимущества динамической структуры данных в виде бинарного дерева по сравнению со списками.
24. Как различаются файлы по способу доступа? Перечислите основные классы для работы с файлами из стандартной библиотеки.

25. Синтаксис оператора, осуществляющего открытие потока и связывание его с файлом. С помощью каких операторов осуществляется вывод данных из файла?
26. Синтаксис оператора, осуществляется ввод данных в файл. Что такое манипуляторы потоков? Какую операцию выполняет манипулятор потока `setw`? Какой манипулятор потока осуществляет форматирование действительных чисел? Каково назначение флагов формата?

Тематика заданий для самостоятельной работы

1. Разработка программной системы для сравнения алгоритмов сортировки данных.
2. Разработка программной системы "Калькулятор" для работы с тригонометрическими функциями.
3. Разработка программной системы для расчета коэффициентов уравнения линейной регрессии.
4. Разработка программной системы для вычисления корней уравнений.
5. Разработка программной системы для решения задачи линейного программирования графическим методом.
6. Обнаружение узлов в сети. Инструмент.
7. Модель разработки Open Source

Оценочные средства, обеспечивающие диагностику сформированности компетенций, заявленных в рабочей программе дисциплины (модуля)

Результат диагностики и сформированности компетенций	Показатели	Критерии	Соответствие/ несоответствие	зачет
ОПК-1	владение базовыми знаниями в области фундаментальных разделов математики в объеме, необходимом для владения математическим аппаратом в гидрометеорологии, для обработки и анализа данных, прогнозирования гидрометеорологических характеристик	<p>Дал грамотный и развернутый ответ на вопросы для подготовки по теоретическим вопросам курса. Выполнены все практические и контрольные работы.</p> <p>Не ответил или ответил неправильно на вопросы для подготовки по теоретическим вопросам курса. Практические и контрольные работы не выполнены.</p>	<p>Соответствие</p> <p>Несоответствие</p>	<p>Зачёт сдан</p> <p>Зачёт не сдан</p>
ОПК-6	способность решать стандартные задачи профессиональной деятельности на основе информационной и	Дал грамотный и развернутый ответ на вопросы для подготовки по теоретическим вопросам курса. Выполнены все практические и контрольные	Соответствие	Зачёт сдан

	<p>библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.</p>	<p>работы.</p> <p>Не ответил или ответил неправильно на вопросы для подготовки по теоретическим вопросам курса. Практические и контрольные работы не выполнены.</p>	<p>Несоответствие</p>	<p>Зачёт не сдан</p>
--	---	---	-----------------------	----------------------

Разработчик:

Най
(подпись)

старший преподаватель
(занимаемая должность)

П.А. Найденов
(инициалы, фамилия)

Программа рассмотрена на заседании кафедры метеорологии и физики околоземного космического пространства

«25» апреля 2019 г. Протокол № 7

и.о.зав. кафедрой Лат И.В. Латышева

Настоящая программа не может быть воспроизведена ни в какой форме без предварительного письменного разрешения кафедры-разработчика программы.