

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ  
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»  
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ  
Кафедра теории упругости и вычислительной математики  
имени академика А.С. Космодамианского

УТВЕРЖДАЮ:

проректор по научно-методической  
и учебной работе

Е.И. Скафа



**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**  
**«ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ»**  
практико-ориентированная дисциплина

Направление подготовки:	<u>01.03.02 Прикладная математика и информатика</u>
Профиль подготовки:	<u>Статистика</u>
Образовательная программа:	<u>Бакалавриат</u>
Квалификация:	<u>Академический бакалавр</u>
Форма обучения:	<u>очная</u>

Донецк 2021



**УТВЕРЖДАЮ:**

Декан факультета математики  
и информационных технологий  
И. А. Моисеенко



подпись

«26» апреля 2021 г.

Рабочая программа учебной дисциплины **«Языки и методы программирования»** составлена на основании Федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 01.03.02 Прикладная математика и информатика, утвержденного приказом Министерства образования и науки Российской Федерации от «10» января 2018г. № 9; Государственного образовательного стандарта высшего образования (ГОС ВО) Донецкой Народной Республики (ДНР) (проекта) по направлению подготовки 01.03.02 Прикладная математика и информатика; Порядка организации учебного процесса в образовательных организациях высшего профессионального образования Донецкой Народной Республики, утвержденного приказом Министерства образования и науки Донецкой Народной Республики от 10.11.2017 г. № 1171 (с изменениями и дополнениями); учебного плана и основной профессиональной образовательной программы высшего образования направления подготовки 01.03.02 Прикладная математика и информатика, профиля: «Прикладная математика и информатика», разработанных в ГОУ ВПО «Донецкий национальный университет».

Разработчик:

Профессор кафедры теории упругости и  
вычислительной математики имени  
академика А.С. Космодамианского,  
д. ф.-м. н., профессор

С.А. Калоеров

Рабочая программа учебной дисциплины утверждена на заседании кафедры теории упругости и вычислительной математики имени академика А.С. Космодамианского

Протокол № 15 от «12» апреля 2021 г.

Заведующий кафедрой

В.И. Сторожев

Рабочая программа учебной дисциплины одобрена учебно-методической комиссией факультета математики и информационных технологий

Протокол № 4 от «14» апреля 2021 г.

Председатель учебно-методической комиссии  
факультета математики и информационных технологий

Л.И. Селякова .

## 1. ОБЛАСТЬ ПРИМЕНЕНИЯ И МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ

Учебная дисциплина «Языки и методы программирования» является практико-ориентированной дисциплиной и относится к базовой части образовательной программы. Для изучения данной учебной дисциплины необходимы знания и умения, формируемые *предшествующими дисциплинами* школьных курсов математики и «Информатика и компьютерные технологии». Знания и умения, полученные в ходе изучения дисциплины «Языки и методы программирования» являются основой для изучения *последующих* дисциплин: «Объектно-ориентированное программирование и стандартная библиотека C++ в численных методах исследования моделей деформирования», «Алгоритмы и структуры данных», «Математические модели и методы теории упругости», «Математические модели деформирования сред с усложненными свойствами»; используется при написании выпускной квалификационной работы.

## 2. СТРУКТУРА ДИСЦИПЛИНЫ

Характеристика учебной дисциплины	Форма обучения	
	Очная	Заочная
Направление подготовки	01.03.02 Прикладная математика и информатика	
Профиль	Статистика	
Образовательная программа	Бакалавриат	
Квалификация	Академический бакалавр	
Количество содержательных модулей и тем	4 (11)	
Дисциплина базовой / вариативной части образовательной программы	Базовой части	
Формы контроля	2 модульных контроля, экзамен в 1-м и 2-м семестрах	
Год подготовки	1	
Семестр	1, 2	
Количество зачетных единиц	11	
Количество часов всего	396	
в т.ч.:		
– лекционных	68	
– практических или семинарских	–	
– лабораторных	136	
– самостоятельной работы	192	
в т.ч. индивидуальное задание	–	
Недельное количество часов	11,6	
в т. ч.: – аудиторных	6	
– самостоятельной работы студента	5,6	

## 3. ОПИСАНИЕ ДИСЦИПЛИНЫ

**Цель изучения дисциплины «Языки и методы программирования»** – с использованием языка C++ обучение студентов методам и приемам проектирования и составления на современных алгоритмических языках многомодульных программ реализации вычислительных процессов, связанных с их будущей профессиональной деятельностью при решении различных задач науки, техники, других областей знаний.

**Задачи:** рассмотрение информации как характеристики объектов реального мира, выработка практических умений поиска, представления и хранения различных видов

информации. Обучение методам составления программ реализации наиболее характерных алгоритмов решения задач направления компьютерно–математического моделирования в прикладной математике и механике деформируемого твердого тела, а также изучения любой информации.

- 1) изучение конструкций языка C++;
- 2) составления программ линейных, разветвляющихся и циклических вычислительных процессов, многомодульных программ;
- 3) обучение методам оптимизации программ всех уровней (разветвляющихся и циклических вычислительных процессов, многомодульных);
- 4) алгоритмизация различных задач вычислительной математики и обработки разнообразной числовой и текстовой информации с составлением многомодульных программ;
- 5) алгоритмизация различных прикладных задач математики, физики и механики с составлением многомодульных программ
- 6) ознакомление с основными элементами объектно-ориентированного программирования.

**Требования к результатам освоения дисциплины.** Процесс изучения дисциплины «Языки и методы программирования» направлен на формирование элементов следующих **компетенций** в соответствии с ФГОС ВО РФ, ГОС ВО ДНР (проект) по направлению подготовки 01.03.02 Прикладная математика и информатика и основной профессиональной образовательной программы высшего образования направления подготовки 01.03.02 Прикладная математика и информатика, профиля: «Статистика»:

<b>Универсальные компетенции (УК):</b>	
Наименование категории (группы) универсальных компетенций: «Системное и критическое мышление»	
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
<b>Общепрофессиональные компетенции (ОПК):</b>	
ОПК-1	Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности
ОПК-2	Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач
<b>Профессиональные компетенции (ПК):</b>	
ПК-1	Способен выполнять научно-исследовательские работы в соответствии с техническим заданием в составе научного коллектива по отдельным разделам темы
ПК-2	Способен проводить обработку и анализ научной информации и результатов исследований

**Индикаторы достижения компетенций и результаты обучения.** Достижение компетенций оценивается на основе таких индикаторов и соответствующих им результатов обучения:

Категории универсальных компетенций	Универсальные компетенции	Индикаторы	Результаты обучения
Системное и	УК-1. Способен	УК-1. И-1	<b>Знает</b> , как осуществлять поиск, выбор, систематизацию, обобщение и

критическое мышление	осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач		критический анализ информации
			<b>Умеет</b> осуществлять поиск, выбор, систематизацию, обобщение и критический анализ информации
		УК-1. И-2	<b>Знает как</b> применять методы системного подхода для решения поставленных задач
			<b>Умеет</b> применять методы системного подхода для решения поставленных задач

Общепрофессиональные компетенции	Индикаторы	Результаты обучения
ОПК-1. Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1. И-1	Знает, как применять основные положения и концепции в области прикладной математики, физики, механики и информатики
		Умеет применять основные положения и концепции прикладной математики, физики, механики и информатики
	ОПК-1. И-2	Знает, как применять основную терминологию прикладной математики и информатики, физики и механики при решении задач профессиональной деятельности
		Умеет применять основную терминологию прикладной математики и информатики, физики и механики при решении задач профессиональной деятельности
ОПК-2. Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	ОПК-2, И-1	Знает, как использовать и адаптировать соответствующие математические методы для разработки алгоритмов решения прикладных задач
		Умеет использовать и адаптировать соответствующие математические методы для разработки алгоритмов решения прикладных задач
	ОПК-2, И-2	Знает, как использовать современные системы программирования для реализации алгоритмов решения различных задач прикладной математики и механики
		Умеет использовать современные системы программирования для реализации алгоритмов решения различных задач прикладной математики и механики

Профессиональные компетенции	Индикаторы	Результаты обучения
ПК-1. Способен выполнять научно-исследовательские работы в соответствии с техническим заданием в составе	ПК-1, ИПК-3	Знает, как на основе анализа получаемых результатов численных экспериментов модифицировать методы решения задач, их алгоритмизацию и конкретную реализацию при решении прикладных задач математики, механики и компьютерных наук
		Умеет на основе анализа получаемых результатов численных экспериментов модифицировать методы



научного коллектива по отдельным разделам темы		решения задач, их алгоритмизацию и конкретную реализацию при решении прикладных задач математики, механики и компьютерных наук
ПК-2. Способен выполнять научно-исследовательские работы в соответствии с техническим заданием в составе научного коллектива по отдельным разделам темы	ПК-2, ИПК-2	Знает, как оформлять результаты научно-исследовательских работ и вычислительных экспериментов в соответствии с существующими стандартами
		Умеет оформлять результаты научно-исследовательских работ и вычислительных экспериментов в соответствии с существующими стандартами

#### 4. ФОРМЫ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Дисциплина «Языки и методы программирования» предусматривает следующие формы организации учебного процесса: лекции, лабораторные занятия, самостоятельную работу студентов.

Материал излагается с использованием объяснительно-иллюстративных, эвристических и исследовательских методов преподавания. При проведении лекций и лабораторных занятий используются мультимедийные презентации, раздаточные материалы.

В учебном процессе широко применяются активные и интерактивные формы проведения занятий (разбор конкретных ситуаций, дискуссия, полемика), внеаудиторная самостоятельная работа, балльно-рейтинговая система оценки успеваемости, личностно-ориентированное обучение, проблемное обучение. В учебном процессе используются интернет-ресурсы по данному курсу; рассматриваются задачи, максимально приближенные к конкретным практическим ситуациям, тесты, самостоятельная работа; контрольные работы.

Самостоятельная работа студентов предусматривает выполнение индивидуальных заданий, подготовку к лабораторным занятиям, изучение учебно-методической литературы, составление конспектов, подготовку презентаций и докладов.

##### Тематический план «Языки и методы программирования»

Темы	Вопросы темы
<b>Содержательный модуль 1. Программирование простейших алгоритмов</b>	
<b>1. Введение в программирование</b>	1.1. Введение в программирование. Применение ЭВМ, краткая история развития ЭВМ и программного обеспечения, арифметические основы АЦВМ, их устройства и функционирование. 1.2. Алгоритм, блок-схемная и операторная запись алгоритмов, 1.3. Примеры блок-схемного программирования линейных и разветвляющихся вычислительных процессов. 1.4. Алгоритмические языки и их классификация (машинный язык, ССК, автокоды, машинно-независимые), их общая характеристика. Исходный, объектный и загрузочный модули.
<b>2. Алгоритмический язык C++ и его простейшие</b>	2.1. Алгоритмический язык C++. Символы и основные лексемы языка. Структура программы. Типы данных, их представления в памяти.

<b>конструкции*</b>	2.2. Константы, переменные, массивы базовых типов, указатели. 2.3. Выражения, операнды, операции, их содержание и особенности выполнения.
<b>3. Программирование линейных и разветвляющихся вычислительных процессов*</b>	3.1. Программирование линейных и разветвляющихся процессов. Операторы, препроцессор, стандартные ввод-вывод, 3.2. Оператор выражения, операторы условного, безусловного переходов и множественного ветвления. 3.3. Составление программ линейных и разветвляющихся вычислительных процессов.
<b>4. Программирование простых циклических вычислительных процессов*</b>	4.1. Программирование циклических вычислительных процессов. Понятие цикла. Структура простого цикла. 4.2. Операторы цикла и их использование при составлении программ. 4.3. Операции распределения и освобождения памяти. Работа с одномерным динамическим массивом. 4.4. Методика составления программ простых циклов.
<b>Содержательный модуль 2. Программирование вложенных циклов и построение простейших программ модульной структуры</b>	
<b>5. Вложенные циклы*</b>	5.1. Вложенные циклы, их блок-схемы, формирование понятия вспомогательных переменных во вложенных циклах. Примеры составления программ вложенных циклов. 5.2. Методика составления программ вложенных циклов. Основные переменные, методика их выбора, нахождение вспомогательных переменных, рекуррентных формул. 5.3. Оптимизация в циклах.
<b>6. Функции, построение простейших многомодульных программ*</b>	6.1. Функции, аргументы функции, способы передачи информации в функцию и из функции. Формальные и фактические параметры, прототипы. Составление программ с функциями в виде простых переменных. 6.2. Формальные массивы и формальные функции. Составление программ с их использованием. 6.3. Методика составления простейших многомодульных программ, прототипы.
<b>Содержательный модуль 3. Программирование вычислительных процессов с обработкой одномерных и двумерных массивов</b>	
<b>7. Проектирование и составление многомодульных программ для формирования и обработки элементов одномерных массивов*</b>	7.1. Связь между формальными и фактическими параметрами в общем случае. 7.2. Методика проектирования и составления многомодульных программ. 7.3. Алгоритмизация вычислительных процессов для работы с одномерными массивами по их обработке (формирование элементов, нахождения максимальных или минимальных элементов или их координат, их расположения в порядке возрастания или убывания, удаления или вставки).
<b>8. Проектирование и составление многомодульных программ для формирования и обработки элементов</b>	Алгоритмизация вычислительных процессов для работы с многомерными массивами по их обработке (формирование строк и столбцов двумерной матрицы, нахождения максимальных или минимальных элементов или их координат, расположения элементов их строк или столбцов в порядке возрастания или убывания их элементов).

<b>многомерных массивов*</b>	
<b>9. Проектирование и составление многомодульных программ реализации некоторых численных методов*</b>	Составление многомодульных программ для организации некоторых вычислительных процессов (вычисление определенных интегралов и решение трансцендентных уравнений)
<b>Содержательный модуль 4. Программирование сложных вычислительных процессов</b>	
<b>10. Программирование сложных вычислительных процессов*</b>	10.1 Функция main, перегрузка функций. 10.2. Шаблонные функции. Общая методика проектирования и составления многомодульных программ. Рекурсии функций. 10.3. Программирование вычислительных процессов с решением систем линейных алгебраических уравнений и вычислением значений различных функций, нахождения экстремальных точек функции с реализацией сложных рекуррентных соотношений между величинами
<b>11. Введение в объектно-ориентированное программирование*</b>	11.1. Введение в объектно-ориентированное программирование. 11.2. Классы, члены класса, секции доступа. Объекты, операции над ними. 11.3. Описание и использование функций-членов класса. 11.4. Конструкторы и деструкторы классов, их описание, определение и использование. 11.5. Дружественные функции, их использование. 11.6 Перегрузка операций. 11.7. Понятие о производных классах, описание доступа. 11.8. Форматный потоковый ввод-вывод.

\* – практико-ориентированные темы.

#### Структура дисциплины «Языки и методы программирования» по видам учебной деятельности

Названия содержательных модулей и тем	Количество часов								
	Очная форма обучения					Заочная форма обучения			
	Всего	В Т.Ч.				е	В Т.Ч.		
		Лекции	Практические	Лабораторные	Самостоятельная работа		Лекции	Практические	Лабораторные
<b>Содержательный модуль 1. Программирование простейших алгоритмов</b>									
1. Введение в программирование	16	6		2	8	–			
2. Алгоритмический язык C++ и его простейшие конструкции	20	6		4	10	–			
3. Программирование линейных и разветвляющихся процессов	22	2		8	12	–			



4. Программирование простых циклических вычислительных процессов	46	4		20	22	–				
<b>Итого по содержательному модулю 1</b>	<b>104</b>	<b>18</b>		<b>34</b>	<b>52</b>	<b>–</b>				
<b>Содержательный модуль 2. Программирование вложенных циклов и построение простейших программ модульной структуры</b>										
5. Вложенные циклы	56	8		18	30					
6. Функции, построение простейших многомодульных программ	58	10		20	28					
<b>Итого по содержательному модулю 2</b>	<b>114</b>	<b>18</b>		<b>38</b>	<b>58</b>					
<b>Содержательный модуль 3. Программирование вычислительных процессов с обработкой одномерных и двумерных массивов</b>										
7. Проектирование и составление многомодульных программ для формирования и обработки элементов одномерных массивов *	42	6		16	20					
8. Проектирование и составление многомодульных программ для формирования и обработки элементов двумерных массивов*	42	6		16	20					
9. Проектирование и составление многомодульных программ реализации некоторых численных методов*	24	4		10	10					
<b>Итого по содержательному модулю 3</b>	<b>108</b>	<b>16</b>		<b>42</b>	<b>50</b>					
<b>Содержательный модуль 4. Программирование сложных вычислительных процессов</b>										
10. Программирование сложных вычислительных процессов*	52	8		22	22					
11. Введение в объектно-ориентированное программирование*	18	8			10					
<b>Итого по содержательному модулю 4</b>	<b>70</b>	<b>16</b>		<b>22</b>	<b>32</b>					
<b>Всего</b>	<b>396</b>	<b>68</b>		<b>136</b>	<b>192</b>					

## 5. ТЕМАТИКА ЛЕКЦИОННЫХ, ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ ЗАНЯТИЙ

### Темы лекционных занятий

№ п/п	Название темы	Количество часов	
		Очная форма	Заочная форма
	<b>Содержательный модуль 1</b>		
1	Введение в программирование	6	
2	Алгоритмический язык C++ и его простейшие конструкции	6	
3	Программирование линейных и разветвляющихся вычислительных процессов	2	
4	Программирование простых циклических вычислительных	4	

	процессов		
	<b>Итого по содержательному модулю 1</b>	<b>18</b>	
	<b>Содержательный модуль 2</b>		
5	Вложенные циклы	8	
6	Функции, построение простейших многомодульных программ	10	
	<b>Итого по содержательному модулю 2</b>	<b>18</b>	
	<b>Содержательный модуль 3</b>		
7	Проектирование и составление многомодульных программ для формирования и обработки элементов одномерных массивов	6	
8	Проектирование и составление многомодульных программ для формирования и обработки элементов многомерных массивов	6	
9	Проектирование и составление многомодульных программ для реализации некоторых численных методов (вычисления определенных интегралов и решения трансцендентных уравнений)	4	
	<b>Итого по содержательному модулю 3</b>	<b>16</b>	
	<b>Содержательный модуль 4</b>		
10	Программирование сложных вычислительных процессов	8	
11	Введение в объектно-ориентированное программирование	8	
	<b>Итого по содержательному модулю 4</b>	<b>16</b>	
	<b>Всего</b>	<b>68</b>	

Тексты лекций приведены в учебном пособии: Калоеров С. А. Программирование на языке C++ : учеб. пособие / С. А. Калоеров ; Донецкий нац. ун-т. - Изд. 3-е. – Донецк : Юго-Восток, 2009. – 298 с.

#### Темы лабораторных работ

№ п/п	Название темы	Количество часов	
		Очная форма	Заочная форма
	<b>Содержательный модуль 1</b>	<b>34</b>	
1	Изучение интегрированной среды Visual Studio	2	
2	Алгоритмический язык C++ и его простейшие конструкции (составление программ на выражения и блок-схемы)	4	
3	Программирование линейных и разветвляющихся вычислительных процессов	8 (2, КР1)	
4	Программирование простых циклических вычислительных процессов	20 (4, МКР1)	
	<b>Содержательный модуль 2</b>	<b>38</b>	
5	Составление программ с вложенными циклами	18(4, КР3)	
6	Функции, построение простейших многомодульных программ	20((2, МКР2)	
	<b>Содержательный модуль 3</b>	<b>42</b>	
7	Проектирование и составление многомодульных программ для формирования и обработки элементов одномерных массивов (формирование элементов, нахождения максимальных или минимальных элементов или их координат, их расположения в порядке возрастания или убывания, удаления или вставки)	16 (4, КР6)	
8	Проектирование и составление многомодульных программ для формирования и обработки элементов многомерных	16 (4, МК3)	

	массивов (формирование строк и столбцов двумерной матрицы, нахождения максимальных или минимальных элементов или их координат, расположения элементов их строк или столбцов в порядке возрастания или убывания их элементов)		
9	Проектирование и составление многомодульных программ для реализации некоторых численных методов (вычисления определенных интегралов и решения трансцендентных уравнений)	10 (1, КР5)	
	<b>Содержательный модуль 4</b>	<b>22</b>	
10	Составление общих многомодульных программ (формирование матриц и решение систем линейных уравнений, вычисление значений функций в точках)	4(6, МК4)	
	<b>Всего</b>	<b>136</b>	

Содержание лабораторных работ и методические рекомендации к их выполнению приведены в учебном пособии «Калоеров С. А. Программирование на языке С++ : учеб. пособие / С. А. Калоеров ; Донецкий нац. ун-т. - Изд. 3-е. – Донецк : Юго-Восток, 2009. – 298 с.» и методическом пособии «Методические указания и задания к лабораторным занятиям по программированию : для студентов специальности "Прикладная математика и информатика" / Сост. С. А. Калоеров, Е. В. Авдюшина, А. И. Занько, М. В. Фоменко, Л. Н. Шкодина, ; Донец. нац. ун-т. – Донецк : ДонНУ, 2018. – 104 с.»

## 6. ОРГАНИЗАЦИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

№ п/п	Название темы	Количество часов	
		Очная форма	Заочная форма
	<b>Содержательный модуль 1</b>	<b>52</b>	
1	Введение в программирование	8	
2	Алгоритмический язык С++ и его простейшие конструкции	10	
3	Программирование линейных и разветвляющихся вычислительных процессов	12	
4	Программирование простых циклических вычислительных процессов	22	
	<b>Содержательный модуль 2</b>	<b>58</b>	
5	Вложенные циклы	30	
6	Функции, построение простейших многомодульных программ	28	
	<b>Содержательный модуль 3</b>	<b>50</b>	
7	Проектирование и составление многомодульных программ для формирования и обработки элементов одномерных массивов	20	
8	Проектирование и составление многомодульных программ для формирования и обработки элементов многомерных массивов	20	
9	Проектирование и составление многомодульных программ для реализации некоторых численных методов (вычисления определенных интегралов и решения трансцендентных уравнений)	10	
	<b>Содержательный модуль 4</b>	<b>32</b>	
10	Программирование сложных вычислительных процессов*	22	
11	Введение в объектно-ориентированное программирование*	10	
	<b>Всего</b>	<b>192</b>	

Содержание самостоятельной работы по темам и методические рекомендации по ее выполнению приведены в учебном пособии «Калоеров С. А. Программирование на языке C++ : учеб. пособие / С. А. Калоеров ; Донецкий нац. ун-т. - Изд. 3-е. – Донецк : Юго-Восток, 2009. – 298 с.»

## 7. КОНТРОЛЬНЫЕ ВОПРОСЫ К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

### Содержательный модуль 1. Программирование простейших алгоритмов

1. Назовите основные символы языка C++.
2. Что такое управляющие символы или эскейп-последовательности? Какие эскейп-последовательности Вы знаете и для чего они используются?
3. Что Вы понимаете под основными лексемами? Назовите основные лексемы программ.
4. Что входит в "пробельные символы"?
5. Где и как записываются комментарии? Какая форма записи комментариев и когда предпочтительнее?
6. Какие типы величин приняты в языке C++? Какие спецификаторы применяются для обозначения величин базовых типов?
7. Как в памяти компьютера представляются величины типов целый и с плавающей точкой с различными спецификаторами типа?
8. В каких пределах на Вашем компьютере изменяются значения величин типов целый и с плавающей точкой?
9. Чем отличаются друг от друга представления в памяти целого числа 5 и числа 5 с плавающей точкой?
10. Почему в арифметических операциях перед их выполнением операнды должны приводиться к одному типу?
11. Чему равны значения следующих выражений:
12.  $3.4 + \text{int } 3.9$ ;  $5 / 2$ ;  $5 \% 2$ ;  $x = 2$ ;  $x = 2, 3$ ;  $x = (2, 3)$ ?
13. В каких операциях и в каких формах могут приводиться указатели? Что участвует в операции для различных форм?
14. Какого типа должны быть значения индексных выражений?
15. В каких операциях может участвовать имя массива и имя массива с предшествующей звездочкой?
16. Пусть  $x$  – имя массива,  $p$  – указатель. Что означают выражения
17.  $p = \&x[2]$ ;  $p++$ ;  $*x + *(p+4) + x[12]$ ?
18. Можно ли писать
19.  $++x$ ;  $++p$ ;  $++p+1$ ;  $++(p+1)$ ;  $++*(p+1)$ ;  $*++(p+1)$ ?
20. Как определяется тип значения выражения?
21. Всегда ли одинаковы значения выражений  $k*m/n$  и  $k*(m/n)$ ?
22. В каких операциях могут участвовать константы и переменные перечисляемого типа?
23. Всегда ли с помощью скобок можно менять порядок выполнения операций выражения? Каким образом изменять порядок выполнения операций, если не помогают скобки?
24. Записать, что  $i$  не равно  $m$  и  $n$ ; лежит между 1 и 9 и не равно 5.
25. Что является отличительным признаком оператора?
26. Что подразумевается под составным оператором или блоком?
27. Для чего нужны метки и как они записываются?
28. Может ли в функции  $m1$  быть именем и переменной, и метки?
29. Могут ли в разных функциях использоваться одинаковые переменные или одинаковые метки?
30. Как записываются комментарии?



31. Как выполняются операторы if и if-else, оператор switch?
32. Пусть алгоритм состоит из двух ветвей. Возможности какого из операторов if-else или switch шире для реализации этого алгоритма?
33. Какую роль в операторе switch играет метка default ?
34. Какой из операторов if-else или switch следует использовать, когда анализируется принадлежность значения промежутку и когда анализируется его совпадение с одной из дискретных констант?
35. Что называется циклом?
36. Какие величины называются переменными цикла?
37. Какие переменные цикла называются рекуррируемыми, а какие не рекуррируемыми?
38. Какие переменные цикла следует подготовить и изменять в цикле, а какие нет?
39. Какими блок-схемами реализуются циклические вычислительные процессы? Каково назначение каждого из этих блоков?
40. Каким образом начинаются, повторяются и завершаются циклы в случае блок-схем 3.1–3.3?
41. Почему нежелательно блок изменения переменных приводить перед арифметическим блоком?
42. Какие операторы цикла существуют и как они исполняются?
43. Каким оператором цикла лучше программировать те или другие циклические вычислительные алгоритмы?
44. Где при использовании оператора цикла for рекуррируемым переменным цикла присваиваются начальные значения при отсутствии выражения e1?

## Содержательный модуль 2. Программирование вложенных циклов и построение простейших программ модульной структуры

45. Приведите блок-схему двойного цикла.
46. Пусть в двойном цикле внешний цикл по k, внутренний по – p. Изобразите, где нужно изменять переменные: а) зависящие только от k, б) зависящие только от p или от k и p совместно?
47. Нарисуйте блок-схему тройного цикла.
48. Найти все натуральные числа, не превосходящие n и делящиеся на каждую из своих цифр.
49. Расположить элементы массива x[n] в порядке возрастания (убывания) модулей.
50. Вычислить произведение матриц  $\|c_{ij}\| = \|a_{ik}\| \|x\| \|b_{kj}\|$ , если его элементы находятся по формуле
 
$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj} \quad (i = \overline{1, m}; \quad j = \overline{1, n}).$$
51. Вычислить сумму произведений каждого элемента массива x[n] на каждый его последующий элемент.
52. Вычислить сумму элементов матрицы x[n][n], которые по модулю меньше 1.
53. Что такое программный модуль (функция), его аргументы?
54. Могут ли в разных функциях встречаться одни и те же метки, одни и те же переменные?
55. Как вы понимаете аргументы вычислительного процесса функции, как они передаются в функцию?
56. Какие имена называются внешними?
57. Объясните назначение формальных и фактических параметров.
58. Каким образом данные передаются в функцию и из нее?
59. Что является признаком головной функции?
60. Как определяются и как вызываются функции?

### Содержательный модуль 3. Программирование вычислительных процессов с обработкой одномерных и двумерных массивов

61. Что характеризует и как записывается класс памяти функции?
62. Можно ли при описании формальных параметров последовательные однородные формальные параметры снабдить одним спецификатором типа?
63. Как информация передается в функцию и из нее?
64. Как составляется проект многомодульной программы задачи?
65. Как Вы понимаете аргументы модулей-функций и аргументы всей задачи?
66. Чем могут отличаться формальные параметры функции от ее аргументов?
67. Что такое фактические параметры вызова функции? Чем они могут задаваться?
68. Какими способами можно передать данные в функцию и из нее? Как эти передачи происходят при входе в функцию и выходе из нее?
69. Какие ограничения накладываются на формальные массивы? Могут ли формальный и фактический массивы отличаться друг от друга по размерностям, объемам, типам элементов?
70. Как заказывать память в свободном поле для переменного массива  $x[n][j]$ ?
71. Какая должна существовать связь между формальными параметрами формальной функции и формальными параметрами функций, вызываемых этой формальной функцией?
72. Что предпочтительнее – передача аргументов через внешние объекты или через связь формальных и фактических параметров?

### Содержательный модуль 4. Программирование сложных вычислительных процессов

73. Среди строк (столбцов) матрицы  $x[n][n]$  найти строку (столбец) с минимальной суммой модулей элементов.
74. Среди элементов матрицы  $x[n][n]$  найти максимальный по модулю. Переставить строки и столбцы таким образом, чтобы этот элемент был первым  $x[0][0]$ .
75. Определить, являются ли линейно независимыми три заданных вектора  $\{a_i\}$ ,  $\{b_i\}$ ,  $\{c_i\}$  ( $i = 1, N$ ).
76. Строки матрицы  $x[n][m]$  переставить в порядке убывания элементов первого столбца.
77. Что называется классом? Как он описывается?
78. Что может быть членами класса и как они описываются?
79. Секции каких уровней доступа могут иметь классы? Как они помечаются? Какие ограничения накладываются на использование членов из секций с различными уровнями доступа?
80. Для чего нужны имена классов?
81. Что такое объект, массив объектов, указатель, ссылка или функция типа класса? Как они описываются?
82. Как описываются внешние и локальные объекты?
83. Опишите массив  $y$  из 10 элементов типа класса  $A_{Class}$ , содержащего переменную  $g$  типа  $int$  и указатель  $p$  типа  $char$ ?
84. Могут ли члены класса быть объектами и функциями типа данного класса?
85. Как описываются классовые и внеклассовые функции программы типа данного класса, с формальными параметрами типа класса?
86. Как осуществляется доступ к членам класса внутри функций-членов?

## 8. ОБРАЗЕЦ ЗАДАНИЯ МОДУЛЬНОГО КОНТРОЛЯ

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»

Образовательная программа: бакалавриат

Направление подготовки: 01.03.02 Прикладная математика и информатика

Профиль: Статистика

Очная форма обучения. Семестр: 2

Учебная дисциплина: Языки и методы программирования

### Модульная контрольная работа

#### Вариант № n

- 1). Как описываются и используются формальные функции? Как согласуются фактические параметры вызова и формальные параметры описания формальной функции?
- 2). Опишите методику проектирования и составления многомодульных программ.
- 3). Что означает шаблон функции, для чего он нужен? Как он записывается и как по нему создаются шаблонные функции?
- 4). Составить программу вычисления значений функции

$$\sigma = 3\varphi(x) + x^2\psi(x)$$

в точках

$$x_k = \cos \frac{k\pi}{12} + d \sin \frac{k\pi}{12} \quad (k = \overline{0, 12}).$$

При этом

$$\varphi(x) = \sum_{k=1}^n \left[ \frac{a_k}{(x+s)^k} + b_k P_k(x) \right], \quad \psi(x) = \sum_{k=1}^n \left[ \frac{b_k}{(x+2s)^k} + 2a_k P_k(x) \right];$$

$$P_0(x) = 1, \quad P_1(x) = x, \quad P_2(x) = x^2 - 2m,$$

$$P_{k+2}(x) = x^2 P_{k+1}(x) + xm P_k(x) + P_{k-1}(x), \quad k = 1, 2, \dots;$$

$a_k, b_k$  – неизвестные постоянные, определяемые из решений систем

$$a_k + \sum_{p=1}^n \frac{(k+p)!}{k! p!} \varepsilon^{k-p} (\varepsilon^p - m^{2k}) \beta_{k+p} a_p = \frac{m^k}{(k+1)!} f_1(k) \quad (k = \overline{1, n}),$$

$$k b_k + \sum_{p=1}^n \frac{k!}{p!} \varepsilon^k m^p (\varepsilon^p - m^2) \beta_{k+p} b_p = \frac{f_2(k)}{(k+1)!} \quad (k = \overline{1, 2n});$$

$$f_1(k) = m^k + k^2 d^3 + c_k^k, \quad f_2(k) = \sin k + 2k \varepsilon^{-k} + m^2 + d^2 + f_1(k);$$

$$\beta_1 = 1, \quad \beta_{j+1} = \frac{2\beta_j}{j+3} + m;$$

$n$  – целое,  $d, m, s, \varepsilon$  – вещественные данные;  $c_k$  – элементы вещественного массива.

Напечатать значения  $\sigma$  во всех точках, а также минимальное значение с указанием точки, в которой оно достигается.

### 9. КРИТЕРИИ ОЦЕНИВАНИЯ ЗАДАНИЯ МОДУЛЬНОГО КОНТРОЛЯ

Номер задания	Количество баллов
1	3
2	3
3	3

<b>Всего</b>	<b>25</b>
--------------	-----------

### 10. ОБРАЗЕЦ ЭКЗАМЕНАЦИОННОГО БИЛЕТА

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»

Образовательная программа: бакалавриат

Направление подготовки: 01.03.02 Прикладная математика и информатика

Профиль: Статистика

Очная форма обучения. Семестр: 2

Учебная дисциплина: Языки и методы программирования

#### Экзаменационный билет № n

1. Арифметические операции, их операнды, результаты, особенности.
2. Вложенный цикл, методика его программирования
3. Составить программу вычисления и вывода элементов массивов  $x_k, y_k, z_k$ , если

$$x_k = 1 + \alpha(k) a_k + \sum_{p=1}^{2n} \frac{(k+1)!}{(k+p)!} \varepsilon^{k+p} (m^k + \varepsilon^p) \gamma_{k+1} \alpha(k+p) \quad (k = \overline{1, 2n}),$$

$$y_k = \varepsilon + \beta(k) a_k + \sum_{p=1}^{2n} \frac{(k+1)!}{(k+p)!} (2\varepsilon)^{k-p} (m^k + \varepsilon^{2p}) \gamma_{k+1} \alpha(k+p) \quad (k = \overline{1, 2n}),$$

$$z_k = \delta(k) a_k + \sum_{p=1}^n \frac{(k+1)!}{(k+p)!} (1 + 2\varepsilon)^k \varepsilon^{k+p} (m^{2k} + \varepsilon^p) \gamma_{k+1} \alpha(k+p) \quad (k = \overline{1, n}).$$

Здесь

$$\gamma_1 = 10^{-7}, \quad \gamma_{r+1} = r \gamma_r + \sqrt{m^2 + 2};$$

$$\alpha(k) = k^2 + m^3 + 2 + a_k; \quad \beta(k) = k \sin(k) + m^2; \quad \delta(k) = \varepsilon^k + mk;$$

$n$  – целое,  $\varepsilon, m$  – вещественные исходные данные;  $a_k$  – элементы вещественного исходного массива.

Утверждено на заседании кафедры теории упругости и вычислительной математики имени академика А.С.Космодамианского, протокол № \_\_\_\_\_ от “\_\_\_” \_\_\_\_\_ 20\_\_ г.

Заведующий кафедрой

\_\_\_\_\_ В.И.Сторожев

Экзаменатор

\_\_\_\_\_ С.А.Калоеров

### 11. КРИТЕРИИ ОЦЕНИВАНИЯ ЭКЗАМЕНАЦИОННОГО ЗАДАНИЯ

Номер задания	Количество баллов
1	10
2	10
3	60
20% от результатов работы в семестре	20
<b>Всего</b>	<b>100</b>



## 12. КРИТЕРИИ ОЦЕНИВАНИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Самостоятельная работа (включая выполнение СРС) оценивается в 25 баллов. В разрезе отдельных тем оценивание осуществляется следующим образом.

### Оценивание СРС и ИРС по дисциплине «Языки и методы программирования»

Те ма	Названия содержательных модулей и тем	СРС	ИРС
	<b>Содержательный модуль 1</b>		
1	Введение в программирование		
2	Алгоритмический язык C++ и его простейшие конструкции	1	
3	Программирование линейных и разветвляющихся вычислительных процессов	1	
4	Программирование простых циклических вычислительных процессов	2	
	<b>Итого по 1-му содержательному модулю</b>	3	
	<b>Содержательный модуль 2</b>		
5	Вложенные циклы	1	
6	Функции, построение простейших многомодульных программ	1	
	<b>Итого по 2-му содержательному модулю</b>	2	
	<b>Содержательный модуль 3</b>		
7	Проектирование и составление многомодульных программ для формирования и обработки элементов одномерных массивов	1	
8	Проектирование и составление многомодульных программ для формирования и обработки элементов многомерных массивов	1	
9	Проектирование и составление многомодульных программ для реализации некоторых численных методов (вычисления определенных интегралов и решения трансцендентных уравнений)	1	
	<b>Итого по 3-му содержательному модулю</b>	3	
	<b>Содержательный модуль 4</b>		
10	Программирование сложных вычислительных процессов*	1	
11	Введение в объектно-ориентированное программирование*	1	
	<b>Итого по 4-му содержательному модулю</b>	2	
	<b>Всего</b>	<b>10</b>	

## 13. КРИТЕРИИ ОЦЕНИВАНИЯ ОБЩЕЙ УСПЕВАЕМОСТИ

Общая оценка знаний студентов по дисциплине проводится по 100-балльной шкале согласно таким критериям, приведенным в таблице ниже. *Организационно-учебная работа студента* в аудитории оценивается на основе таких критериев как посещаемость занятий, активность во время проведения лекционных и лабораторных занятий (вопросы лектору по теме лекционного материала, участие в обсуждении пройденного материала, решение задач и ситуаций у доски и т.п.).

Зачетные модули	№ п.п	Виды контрольных мероприятий	Количество баллов
Содержательный модуль 1	1.	Индивидуальные задания	10
	2.	Контрольные работы	15
	3.	МКР1	25
Содержательный модуль 2	4.	Индивидуальные задания	10
	5.	Контрольные работы	15

	6.	МКР2	<b>25</b>
<b>Итог за семестр 1</b>			<b>100</b>
Содержательный модуль 3	7	Индивидуальные задания	<b>10</b>
	8	Контрольные работы	<b>15</b>
	9	МКР3	<b>25</b>
Содержательный модуль 4	10	Индивидуальные задания	<b>10</b>
	11	Контрольные работы	<b>15</b>
	12	МКР4	<b>25</b>
<b>Итог за семестр 2</b>			<b>100</b>

#### Порядок оценивания учебных достижений обучающихся

Оценка по шкале ECTS	Оценка по 100-балльной шкале	Оценка по государственной шкале	
		экзамен, дифференцированный зачет	зачет
A	90-100	5 (отлично)	зачтено
B	80-89	4 (хорошо)	зачтено
C	75-79	4 (хорошо)	зачтено
D	70-74	3 (удовлетворительно)	зачтено
E	60-69	3 (удовлетворительно)	зачтено
FX	35-59	2 (неудовлетворительно) с возможностью повторной аттестации	не зачтено
F	0-34	2 (неудовлетворительно) с возможностью повторной сдачи при условии обязательного набора дополнительных баллов	не зачтено

### 14. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОГО ПРОЦЕССА

Учебные занятия проводятся в главном корпусе ДонНУ (83001, г. Донецк, пр. Гурова, 6). Для проведения лекционных и лабораторных занятий требуется аудитория, оборудованная меловой или маркерной доской, мультимедийный проектор и экран, ноутбук, комплект учебной мебели для студентов, рабочее место преподавателя. Выход в Интернет проводной или с использованием Wi-Fi.

Для самостоятельной работы используются текстовые и электронные ресурсы Научной библиотеки университета и других электронных библиотечных баз данных, учебно-методическое обеспечение, представленное в учебно-методических кабинетах главного корпуса (ауд. 604), материально-техническую базу учебной лаборатории «Сетевых компьютерных технологий» (ауд. 606) и учебной лаборатории «Интегрированных сред программирования» (ауд. 610) кафедры теории упругости и вычислительной математики имени академика А.С. Космодамианского.

В процессе обучения студенты имеют возможность использовать учебные материалы по дисциплине «Языки и методы программирования», приведенные в учебном пособии «Калоеров С. А. Программирование на языке C++ : учеб. пособие / С. А. Калоеров ; Донецкий нац. ун-т. - Изд. 3-е. – Донецк : Юго-Восток, 2009. – 298 с.» и в методическом пособии «Методические указания и задания к лабораторным занятиям по программированию : для студентов специальности "Прикладная математика и информатика" / Сост. С. А. Калоеров, Е. В. Авдюшина, А. И. Занько, М. В. Фоменко, Л. Н. Шкодина, ; Донец. нац. ун-т. – Донецк : ДонНУ, 2018. – 104 с.».

### 15. РЕКОМЕНДОВАННАЯ ЛИТЕРАТУРА

№	Наименование	Кол-во	Наличие
---	--------------	--------	---------

п/п		экземпляров в библиотеке ДонНУ	электронной версии в ЭБС
<b>Основная литература</b>			
1.	Калоеров С. А. Программирование на языке С++ : учеб. пособие / С. А. Калоеров ; Донецкий нац. ун-т. - Изд. 3-е. – Донецк : Юго-Восток, 2009. – 298 с. 2-е изд. – Донецк : Юго-Восток, 2004. – 237 с. . 1-е изд Донецк : Юго-Восток, 2002. – 224 с.	50	+
2.	Страуструп Б. Язык программирования С++ / Бьерн Страуструп ; пер. с англ. С. Анисимова, М. Кононова ; под ред. Ф. Андреева, А. Ушакова. – спец. изд. – М. : Бином-Пресс, 2008. – 1098 с. Места выдачи: АНЛ (4)	4	–
3.	Шилдт Г. Самоучитель С++ : [Для программистов и опыт. пользователей] / Герберт Шилдт ; [Пер. с англ. Алексей Жданов]. – 3-е изд. – СПб. и др. : ВНУ-Санкт-Петербург, 2001. – 683 с.	1	–
4.	Методические указания и задания к лабораторным занятиям по программированию : для студентов специальности "Прикладная математика и информатика" / Сост. С. А. Калоеров, Е. В. Авдюшина, А. И. Занько, М. В. Фоменко, Л. Н. Шкодина, ; Донец. нац. ун-т. – Донецк : ДонНУ, 2018. – 104 с.	10	+
<b>Дополнительная литература</b>			
5.	Павловская Т. А. С/С++ : Программирование на языке высокого уровня / Т. А. Павловская. – М. и др. : Питер, 2004. – 461 с. Изд 2-е – Питер, 2008. – 461 с.	3	–
6.	Павловская, Т. А. С/С++. Программирование на языке высокого уровня : учеб. для студентов вузов, обучающихся по направлению "Информатика и вычислит. техника" / Т. А. Павловская. – М. [и др.] : Питер, 2009. – 460 с. Изд 2-е . – М. [и др.] : Питер, 2010. – 460 с.	2	–
7.	Подбельский В.В. Язык Си++. – М.: Финансы и статистика, 1996. – 560 с.	1	–
8.	Евсеев Г. А. Новейший самоучитель работы на компьютере : Эффектив. курс / Г. Евсеев, В. Мураховский, С. Симонович ; Под ред. С. Симоновича. – М. : ДЕСС, 2004. – 688 с.	1	–
9.	Пол А. Объектно-ориентированное програм мирование на С++ / Айра Пол ; Пер. с англ. Д. Ковальчука. – 2-е изд. – М. : БИНОМ ; СПб. : Невский диалект, 1999. – 464 с.	1	–
10.	Сван Т. Освоение Borland С++ 4.5. Практический курс. К.: Диалектика, 1996.– 544 с.	2	–
11.	Фейсон, Тэд. Объективно-ориентированное програм мирование на Borland С++ 4.5 : Руководство для опытных программистов / Пер. с англ. И. Е. Онищенко, О. А. Меженного. – 4-е изд. – К. : Диалектика, 1996. – 541 с.	1	–
12.	Абрамов С. А. Задачи по программированию / С. А.	1	–

	Абрамов, Г. Г. Гнездилова, Е. Н. Капустина, М. И. Седюн. – М. : Наука, 1988. – 224 с.		
13.	Фигурнов, В. Э. IBM PC для пользователя. – 2-е изд. – М. : Финансы и статистика, 1991. – 288 с. 7-е изд. – М. : ИНФРА-М, 1997. – 640 с.	2	–

*Допускается использование ЭБС, с которыми у Университета заключен договор и к которым есть доступ через сайт научной библиотеки ДонНУ со страницы <http://library.donnu.ru/russ/infpro.html>*

## 16. ИНФОРМАЦИОННЫЕ РЕСУРСЫ

<http://www.jip.ru/Scope.htm> - Журнал «Информационные процессы»  
<http://novtex.ru/IT/arhiv.htm> - Журнал «Информационные технологии»  
<http://crm.ics.org.ru/> - Журнал «Компьютерные исследования и моделирование»  
<http://mais-journal.ru/jour> - Журнал «Моделирование и анализ информационных систем»  
<https://support.office.com/ru-ru/word> - справочник по поддержке Microsoft Office  
<http://mondnr.ru/> – Министерство образования и науки Донецкой Народной республики  
<https://www.donippo.org/> – ГОУ ДПО «Донецкий республиканский институт дополнительного педагогического образования»  
<http://ippo-vm.at.ua/> – Отдел математики Донецкого РИДПО  
<http://resobrnadzor.ru/> –Республиканская служба по контролю и надзору в сфере образования и науки

## 17. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

1. Windows 7 PRO (корпоративная лицензия ДОННУ № 46484614);
2. Microsoft Office (корпоративная лицензия ДОННУ лицензия № 46472919);
3. Microsoft Visual Studio (лицензия программы DreamSpark для высших учебных заведений);
4. Лицензии GPL для свободного программного обеспечения: Антивирус Касперского, Libre Office, Adobe Acrobat Reader, xPDF, Paint.NET.