

ГОУ ВПО «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 Е.И. Скафа
«22» апреля 2020 г.

МП

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
«ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ»

Направление подготовки:	01.03.02 Прикладная математика и информатика
Образовательная программа:	бакалавриат
Квалификация:	Академический бакалавр
Форма обучения:	<u>очная, очно-заочная, заочная, в том числе с ускоренным сроком обучения</u> нужное подчеркнуть

Донецк 2020

УТВЕРЖДАЮ:

Декан факультета математики
и информационных технологий
И. А. Моисеенко

«16» апреля 2020

МП

Программа учебной дисциплины «Языки и методы программирования» составлена на основании Государственного образовательного стандарта высшего профессионального образования (ГОС ВПО) Донецкой Народной Республики (ДНР) по направлению подготовки 01.03.02 Прикладная математика и информатика, утвержденного приказом Министерства образования и науки ДНР от «04» апреля 2016 г. № 280;

Порядка организации учебного процесса в образовательных организациях высшего профессионального образования Донецкой Народной Республики, утвержденного приказом Министерства образования и науки ДНР № 1171 от «10» ноября 2017 г.;

учебного плана и основной образовательной программы высшего профессионального образования направления подготовки 01.03.02 Прикладная математика и информатика, разработанных в ГОУ ВПО «Донецкий национальный университет».

Разработчик:

Профессор кафедры теории упругости и
вычислительной математики имени
академика А.С. Космодамианского



С.А. Калоеров

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

Протокол № 11 от «9» апреля 2020 г.
Заведующий кафедрой



В.И. Сторожев

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

Протокол № 8 от «15» апреля 2020 г.

Председатель учебно-методической
комиссии факультета



Л.И. Селякова

ОБЛАСТЬ ПРИМЕНЕНИЯ И МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ

Учебная дисциплина «Языки и методы программирования» относится к циклу базовой части профессионального блока и состоит из 4 модулей. Содержание дисциплины является логическим продолжением школьного курса математики и информатики и формирует основу дальнейшего освоения дисциплин в области алгоритмизации, современных технологий программирования, разработки и управления информационными системами, математического моделирования в компьютерных системах, а также в задачах прикладной математики и механики деформируемого твердого тела.

1. СТРУКТУРА ДИСЦИПЛИНЫ

<i>Характеристика учебной дисциплины</i>				
Направление подготовки	01.03.02 Прикладная математика и информатика			
Профиль				
Образовательная программа	бакалавриат			
Квалификация	Академический бакалавр			
Количество содержательных модулей	4			
Дисциплина базовой / вариативной части образовательной программы	Базовая часть профессионального блока			
Формы контроля (МК, экзамен, зачет)	Модульный контроль, экзамен			
Показатели	очная форма обучения		заочная форма обучения	
	нормат. срок	ускор. срок	нормат. срок	ускор. срок
Количество зачетных единиц (кредитов)	12	12		
Год подготовки	1	1		
Семестр	1, 2	1, 2		
Количество часов	432	432		
- лекционных	68	68		
- практических, семинарских				
- лабораторных	136	136		
- самостоятельной работы	228	228		
в т.ч. индивидуальное задание				
Недельное количество часов,	12.7	12.7		
в т.ч. аудиторных	6	6		

2. ОПИСАНИЕ ДИСЦИПЛИНЫ

Цели и задачи

Цель – Целью дисциплины является 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, разработка практических умений поиска, представления и хранения различных видов информации. Обучение методам составления программ реализации наиболее характерных алгоритмов решения задач направления компьютерно–математического моделирования в приклад-

ной 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 обработке разнообразной числовой и текстовой информации с составлением многомодульных программ;
- 5) 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 ГОС ВПО ДНР по направлению подготовки 01.03.02 Прикладная математика и информатика и основной образовательной программы высшего профессионального образования направления подготовки 01.03.02 Прикладная математика и информатика:

а) общекультурных (ОК):

- способностью работать в команде, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия (ОК-6);
- способностью к самоорганизации и самообразованию (ОК-7);

б) общепрофессиональных (ОПК):

- способностью использовать базовые знания естественных наук, математики и информатики, основные факты, концепции, принципы теорий, связанных с прикладной математикой и информатикой (ОПК-1);
- способностью к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям (ОПК-3);

в) профессиональных (ПК):

научно-исследовательская деятельность:

- способностью собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям (ПК-1);
- способностью понимать, совершенствовать и применять современный математический аппарат (ПК-2);
- способностью критически переосмысливать накопленный опыт, изменять при необходимости вид и характер своей профессиональной деятельности (ПК-3);

проектная и производственно-технологическая деятельность

способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения (ПК-7).

В результате изучения учебной дисциплины студент должен

Знать:

- типы данных, задания констант и переменных, массивов и их элементов, указателей, структуру задания классов и их членов, конструкторов и деструкторов, дружественных функций, перегрузок операций;
- составление выражений, операторов;
- составление программ линейных, разветвляющихся, циклических вычислительных процессов;

- методы программирования простых и вложенных циклов достаточно большой вложенности;
- методы проектирования и составления многомодульных программ;
- составление алгоритмов решения различных задач вычислительной математики и обработки информации;
- процесс трансляции, редактирования и выполнения программ;
- процесс отладки программ.

Уметь:

- профессионально работать за клавиатурой персонального компьютера;
- работать в интегрированной среде C++ по созданию файлов, их редактированию, запуску программ на счет;
- составлять программы разветвляющихся вычислительных процессов и простых циклов;
- в совершенстве владеть методами подготовки и изменения переменных во вложенных циклах;
- проектировать и составлять многомодульные программы сложных вычислительных процессов;
- проектировать классы, составлять функции класса, конструкторы и деструкторы, дружественные функции;
- перегружать знаки операций, составлять простейшие программы для работы с объектами в случае перегруженных операций;
- составлять программы как вычислительного, так и информационного характера.

Владеть:

- навыками математического описания и разработки алгоритмов решения реальных задач;
- навыками проектирования и составления многомодульных программ сложных вычислительных процессов;
- навыками проведения численных исследований по составленным программам и анализа полученных результатов.

3. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ И ФОРМЫ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА

В рамках изучения дисциплины предусмотрены следующие формы организации учебного процесса: лекции, лабораторные занятия, индивидуальные задания, самостоятельная работа студента.

Лекционные занятия предполагают овладение теоретическими основами дисциплины, лабораторные – овладение методами сбора, хранения и обработки информации; выполнение индивидуальных заданий.

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

Текущий контроль осуществляется путем защиты выполненных лабораторных работ и индивидуальных заданий, модульных контрольных работ по проверке теоретических и практических знаний.

В учебном процессе применяются активные и интерактивные формы проведения занятий, внеаудиторная самостоятельная работа, балльно-рейтинговая система оценки успеваемости, личностно-ориентированное обучение, проблемное обучение.

Материал излагается с использованием объяснительно-иллюстративных, эвристических и исследовательских методов преподавания, широко используются презентации.

Порядковый номер и тема	Краткое содержание темы
Содержательный модуль 1	
Тема 1. Введение в программирование.	Введение в программирование. Применение ЭВМ, краткая история развития ЭВМ и программного обеспечения, арифметические основы АЦВМ, их устройства и функционирование, алгоритм, блок-схемная и операторная записи алгоритмов, примеры блок-схемного программирования линейных и разветвляющихся вычислительных процессов. Алгоритмические языки и их классификация (машинный язык, ССК, автокоды, машинно-независимые), их общая характеристика. Исходный, объектный и загрузочный модули.
Тема 2. Алгоритмический язык С++ и его простейшие конструкции	Алгоритмический язык С++. Символы и основные лексемы языка. Структура программы. Типы данных, их представления в памяти. Константы, переменные, массивы базовых типов, указатели. Выражения, операнды, операции, их содержание и особенности выполнения.
Тема 3. Программирование линейных и разветвляющихся процессов.	Программирование линейных и разветвляющихся процессов. Операторы, препроцессор, стандартные ввод-вывод, оператор выражения, операторы условного, безусловного переходов и множественного ветвления. Составление программ линейных и разветвляющихся вычислительных процессов.
Тема 4. Программирование простых циклических вычислительных процессов.	Программирование циклических вычислительных процессов. Понятие цикла. Структура простого цикла. Операторы цикла и их использование при составлении программ. Операции распределения и освобождения памяти. Работа с одномерным динамическим массивом. Методика составления программ простых циклов.
Содержательный модуль 2	
Тема 5. Вложенные циклы	Вложенные циклы, их блок-схемы, формирование понятия вспомогательных переменных во вложенных циклах. Примеры составления их программ. Основные переменные, методика их выбора, нахождение вспомогательных переменных, рекуррентных формул. Оптимизация в циклах.
Тема 6. Функции, построение простейших многомодульных программ	Функции, построение простейших многомодульных программ. Аргументы функции. Способы передачи информации в функцию и из функции. Формальные и фактические параметры. Прототипы. Формальные и фактические функции, массивы (указатели). Методика проектирования составления простейших многомодульных программ.
Содержательный модуль 3	
Тема 7. Проектирование и составление многомодульных программ для реализации некоторых численных методов	Проектирование и составление многомодульных программ. Связь между формальными и фактическими параметрами. Составление многомодульных программ для организации некоторых вычислительных процессов (вычисление определенных интегралов и решение трансцендентных уравнений)
Тема 8. Проектирование и составление многомодульных программ для формирования и обработ-	Алгоритмизация вычислительных процессов для работы с многомерными массивами и решения задач из различных областей знаний. Составление функций обработки массивов числовой информации, функций с формированием динами-

ки элементов одномерных массивов	ческих матриц линейных алгебраических уравнений, много-модульных программ решения задач с решением систем линейных алгебраических уравнений и вычисления функций, максимальных их значений.
<i>Тема 9. Составление много-модульных программ для формирования и обработки элементов много-мерных массивов</i>	Алгоритмизация вычислительных процессов для работы с многомерными массивами (формирование элементов и их сортировка)
<i>Содержательный модуль 4</i>	
<i>Тема 10. Программирование сложных вычислительных процессов</i>	Функция main, перегрузка функций. Шаблонные функции. Общая методика проектирования и составления много-модульных программ. Рекурсии функций. 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
<i>Тема 11. Введение в объектно-ориентированное программирование.</i>	Введение в объектно-ориентированное программирование. Классы, члены класса, секции доступа. Объекты, операции над ними. Описание и использование функций-членов класса. Конструкторы и деструкторы классов, их описание, определение и использование. Дружественные функции, их использование. Перегрузка операций. Понятие о производных классах, описание доступа. Форматный потоковый ввод-вывод.

[illegible]

4. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРОВЕДЕНИЯ ЛЕКЦИОННЫХ, ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Темы лекционных занятий

<i>№ п/п</i>	<i>Название темы</i>	<i>Количество часов</i>
Тема 1	Введение в программирование.	6
Тема 2.	Алгоритмический язык C++ и его простейшие конструкции	6
Тема 3	Программирование линейных и разветвляющихся вычислительных процессов.	2
Тема 4	Программирование простых циклических вычислительных процессов.	4
	<i>Содержательный модуль 2</i>	
Тема 5.	Вложенные циклы	6
Тема 6.	Функции, построение простейших многомодульных программ	10
	<i>Содержательный модуль 3</i>	
Тема 7	Проектирование и составление многомодульных программ для реализации некоторых численных методов	4
Тема 8	Проектирование и составление многомодульных программ для формирования и обработки элементов одномерных массивов	6
Тема 9	Составление многомодульных программ для формирования и обработки элементов многомерных массивов	6
	<i>Содержательный модуль 4</i>	.
Тема 10.	Программирование сложных вычислительных процессов	8
Тема 11.	Введение в объектно-ориентированное программирование	8
	ВСЕГО	68

Темы лабораторных занятий по выполнению индивидуальных заданий
(названия тем аудиторных лабораторных занятий и их объем совпадают с соответствующими темами лекционных занятий, здесь приведены в скобках)

<i>№ п/п</i>	<i>Название темы</i>	<i>Количество часов</i>
	<i>Первый семестр</i>	
1	Изучение интегрированной среды Visual Studio	2
2	Составление программ на линейные вычислительные процессы	2 (4)
3	Составление программ на разветвляющиеся вычислительные процессы	2(2, КР1)
4	Составление программ по программированию простых циклов	4 (4, МКР1)
5	Составление программ с вложенными циклами	4(4, КР3)
6	Составление простейших многомодульных программ с вычислениями величин	2(2)
7	Формальные параметры в виде указателей и имен функций. Составление программ формирования элементов одномерных массивов	2 (2, МКР2)
	<i>Второй семестр</i>	
1	Вычисление определенных интегралов	2 (1)
2	Решение трансцендентных уравнений	2(1, КР5)
3	Сортировка в одномерных массивах (введение элемента, удаление	4 (4, КР6)

	элемента, сортировка по возрастанию-убыванию)	
4	Двумерные массивы, формирование элементов и их сортировки	4 (4, МКЗ)
5	Общая многомодульная программа (формирование матриц и решение систем линейных уравнений, вычисление значений функций в точках)	4(6, МК4)
	ВСЕГО	68 (68)

5. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Организация самостоятельной работы студентов

<i>№ n/n</i>	<i>Название темы</i>	<i>Количество часов</i>
1	Введение в программирование.	12
2	Алгоритмический язык C++ и его простейшие конструкции	12
3	Программирование линейных и разветвляющихся процессов.	12
4	Программирование циклических вычислительных процессов.	16
5	Вложенные циклы	30
6	Функции, построение простейших многомодульных программ	28
	ВСЕГО	228

6. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Индивидуальное задание не предусмотрено.

7. КОНТРОЛЬНЫЕ ВОПРОСЫ К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

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`?
45. Приведите блок-схему двойного цикла.
46. Пусть в двойном цикле внешний цикл по k , внутренний по r . Изобразите, где нужно изменять переменные: а) зависящие только от k , б) зависящие только от r или от k и r 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 $\|c_{ij}\| = \|a_{ik}\| \cdot \|b_{ik}\|$, если его элементы находятся по

формуле
$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj} \quad (i = \overline{1, m}; j = \overline{1, n})$$
.

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?
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?
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 $\{a_i\}$, $\{b_i\}$, $\{c_i\}$ ($i = \overline{1, N}$).
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 `AClass`, 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 МОДУЛЬНОГО КОНТРОЛЯ

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»

Факультет математики и информационных технологий

Направление подготовки: **01.03.02 Прикладная математика и информатика**

Профиль:

Программа подготовки: **бакалавриат**

Семестр **4**

Учебная дисциплина **Языки и методы программирования**

МОДУЛЬНАЯ КОНТРОЛЬНАЯ РАБОТА № 4

ВАРИАНТ №1

- 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

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

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

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

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

n – целое, d, m, s, ε – вещественные данные; c_k – элементы вещественного массива.

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

Утверждено на заседании кафедрой теории упругости и вычислительной математики имени академика А.С.Космодамианского, протокол № ____ от “__” _____ 20__ г.

Заведующий кафедрой
Преподаватель

В.И.Сторожев
С.А.Калоеров

Критерии оценивания модульного контроля

<i>Номер задания</i>	<i>Количество баллов</i>
1	3
2	3
3	3
4	16
Всего	25

9. ОБРАЗЕЦ ЭКЗАМЕНАЦИОННОГО БИЛЕТА

Теоретические вопросы к экзамену

1. Как описываются и как используются формальные функции?
2. Как согласуются фактические параметры вызова и формальные параметры описания формальной функции?
3. Что может быть формальным параметром функции и соответствующим фактическим параметром вызова?
4. Как происходит передача формальным параметрам информации о фактических параметрах для различных случаев задания формальных параметров?
5. Опишите методику проектирования и составления многомодульных программ.
6. Что означает шаблон функции, для чего он нужен? Как он записывается и как по нему создаются шаблонные функции?
7. Что означает перегрузка функций? Как определяются вызываемая функция среди перегруженных?

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»

Факультет математики и информационных технологий

Направление подготовки: **01.03.02 Прикладная математика и информатика**
 Профиль:
 Программа подготовки: **бакалавриат**
 Семестр: **4**
 Учебная дисциплина: **Языки и методы программирования**

БИЛЕТ №1

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

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

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

n – целое, ε, 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
Экзаменатор

В.И.Сторожев

С.А.Калоеров

Критерии оценивания экзамена

Номер задания	Количество баллов
1	10
2	10
3	60
20% от результатов работы в семестре	20
Всего	баллов

10. ОБРАЗЕЦ ТЕСТОВОГО ЗАДАНИЯ (при наличии)

Тестовое задание не предусмотрено.

11. КРИТЕРИИ ОЦЕНИВАНИЯ

Содержание учебного курса состоит из четырех зачетных модулей. В каждый зачетный модуль входит выполнение индивидуальных заданий, контрольных работ, а также МКР, которая содержит вопросы теории и составление обобщающей программы по разделу.

Оценка знаний студентов проводится по 100-балльной шкале согласно следующим критериям:

Распределение баллов, которые могут получить студенты в процессе изучения дисциплины

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	25
Итог за семестр 1			100
Содержательный модуль 3	7	Индивидуальные задания	10
	8	Контрольные работы	15

	9	МКР3	25
Содержательный модуль 4	10	Индивидуальные задания	10
	11	Контрольные работы	15
	12	МКР4	25
Итог за семестр 2			100

Шкала соответствия баллов национальной шкале

Оценка по шкале ECTS	Оценка по 100-балльной шкале	Оценка по государственной шкале (экзамен, дифференцированный зачет)	Оценка по государственной шкале (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 сдачи	не зачтено
F	0-34	2 (неудовлетворительно) с возможностью повторной сдачи при условии обязательного набора дополнительных баллов	не зачтено

12. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОГО ПРОЦЕССА

Для проведения лекционных и лабораторных занятий требуется аудитория на группу, оборудованная меловой и интерактивной доской, мультимедийным проектором и экраном.

Лабораторные занятия проводятся в компьютерном классе, оборудованном компьютерами с лицензионным программным обеспечением, доступом к сети Интернет, столами, доской. Для обеспечения лабораторных занятий по данному курсу необходимы компьютеры, установленное Microsoft Visual Studio.

13. РЕКОМЕНДОВАННАЯ ЛИТЕРАТУРА

№ п/п	Наименование	Кол-во экземпляров в библиотеке ДонНУ	Наличие электронной версии в ЭБС
Основная литература			
1.	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. 2-е изд. - Донецк : Юго-Восток, 2004. - 237 с. . 1-е изд Донецк : Юго-Восток, 2002. - 224 с.	50	+
2.	Страуструп Б. Язык программирования C++ / 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 C++ : [Для программистов и опыт. пользователей] / Герберт Шилдт ; [Пер. с англ. Алексей Жданов]. - 3-е изд. - СПб. и др. : BHV-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	+
<i>Дополнительная литература</i>			
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. - Москва [и др.] : Питер, 2009. - 460 с. Изд 2-е Москва [и др.] : Питер, 2010. - 460 с.	2	-
7.	Подбельский В.В. Язык Си++.- М.: Финансы и статистика, 1996.– 560 .	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 программирование 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 программирование 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. Абрамов, Г. Г. Гнездилова, Е. Н. Капустина, М. И. Седюн. - М. : Наука, 1988. - 224 с.	1	-
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	-

14. 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

15. 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, Apache, BSD для свободного программного обеспечения: FreeLab, Scilab, R Studio, Python, Eclipse, Free Pascal, Tries Mode, Prolog, Антивирус Касперского, Linux Fedora, Libre Office, Adobe Acrobat Reader, xPDF, Blender, КОМПАС-3D LT, Paint.NET, Gimp.

Рабочая программа рассмотрена и переутверждена на заседании кафедры теории упругости и вычислительной математики имени академика А.С. Космодамианского с изменениями (без изменений) на 20__ год.

Протокол № __ от “__” _____ 20__ г.

Заведующий. кафедрой

_____ В.И. Сторожев

Рабочая программа рассмотрена и переутверждена на заседании кафедры теории упругости и вычислительной математики имени академика А.С. Космодамианского с изменениями (без изменений) на 20__ год.

Протокол № __ от “__” _____ 20__ г.

Заведующий. кафедрой

_____ В.И. Сторожев

Рабочая программа рассмотрена и переутверждена на заседании кафедры теории упругости и вычислительной математики имени академика А.С. Космодамианского с изменениями (без изменений) на 20__ год.

Протокол № __ от “__” _____ 20__ г.

Заведующий. кафедрой

_____ В.И. Сторожев

Рабочая программа рассмотрена и переутверждена на заседании кафедры теории упругости и вычислительной математики имени академика А.С. Космодамианского с изменениями (без изменений) на 20__ год.

Протокол № __ от “__” _____ 20__ г.

Заведующий. кафедрой

_____ В.И. Сторожев

Рабочая программа рассмотрена и переутверждена на заседании кафедры теории упругости и вычислительной математики имени академика А.С. Космодамианского с изменениями (без изменений) на 20__ год.

Протокол № __ от “__” _____ 20__ г.

Заведующий. кафедрой

_____ В.И. Сторожев