

Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Донецкий государственный университет»

Факультет математики и информационных технологий
Кафедра теории упругости и вычислительной математики
имени академика А.С. Космодамианского



УТВЕРЖДАЮ
проректор

Машаров

П.А. Машаров

« 29 » марта 2024 г.

МП

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ

Укрупненная группа направлений
подготовки

Программа высшего образования
Направление подготовки

Профиль подготовки
Квалификация
Форма обучения

01.00.00 Математика и механика

Программа бакалавриата

01.03.02 Прикладная математика и
информатика

Прикладная математика и информатика

Бакалавр

Очная

Рабочая программа адаптирована для лиц
с ограниченными возможностями здоровья и инвалидов

Донецк 2024

Рабочая программа дисциплины «Языки и методы программирования» для обучающихся по направлению подготовки 01.03.02 Прикладная математика и информатика (Профиль: Прикладная математика и информатика), 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 2018 г. № 9 (с изм. и доп.), Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры, утвержденного приказом Министерства науки и высшего образования Российской Федерации от 06 апреля 2021 г. № 245 (с изм. и доп.), в соответствии с учебным планом, утвержденным Ученым советом ФГБОУ ВО «ДонГУ» для набора в 2024 г.

Разработчик:

профессор кафедры теории упругости
и вычислительной математики
им. акад. А.С. Космодамианского,
доктор физ.-мат. наук, профессор



С.А. Калоев

Рабочая программа одобрена на заседании кафедры теории упругости и
вычислительной математики им. акад. А.С. Космодамианского.
Протокол от 26.03.2024 г. № 10

Врио заведующего кафедрой



Р.Н. Нескородев

СОГЛАСОВАНО:

Декан факультета математики и
информационных технологий
28.03.2024 г.



И.А. Моисеенко

Учебно-методическая комиссия факультета математики и информационных технологий.
Протокол от 27.03.2024 г. № 3.
Председатель



Л. И. Селякова

Руководитель основной профессиональной
образовательной программы,
д-р физ.-мат. наук, доцент
26.03.2024 г.



Р.Н. Нескородев

1. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

1.1. Требования к предварительной подготовке обучающихся, предшествующие и сопутствующие дисциплины, на которых основывается изучение данной:

Школьные курсы математики и «Информатика и компьютерные технологии»;

1.2. Дисциплины, курсовые работы и практики, для которых освоение данной дисциплины необходимо как предшествующее:

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 курсовых и выпускной квалификационной работы.

2. ОПИСАНИЕ ДИСЦИПЛИНЫ

2.1. Общая характеристика

Наименование показателя	Значение показателя
Название образовательной программы	01.03.02 Прикладная математика и информатика (Профиль: Прикладная математика и информатика)
Шифр и название в соответствии с учебным планом	Б1.Б.23 Языки и методы программирования
Часть образовательной программы	Базовая часть
Количество зачетных единиц / всего часов	11 / 396

2.2. Распределение часов по формам и периодам обучения

Форма обучения	курс	семестр	Общее количество часов					Форма контроля
			лекционных	лабораторных	практических	самостоятельной работы + контроль	всего	
Очная	1	1	34	68	-	114	216	экзамен
Очная	1	2	30	60	-	90	180	экзамен
Очная, всего			64	128	-	204	396	

3. ЦЕЛИ ДИСЦИПЛИНЫ

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. 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.

4. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ КОМПОНЕНТА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ, ИХ ИНДИКАТОРЫ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

4.1. Компетенции.

ОПК-2. Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач.

ОПК-5. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.

4.2. Индикаторы компетенций

ОПК-2.2. Применяет язык программирования C++ и методы программирования для использования и адаптации существующих математических методов и систем программирования для разработки и реализации алгоритмов решения прикладных задач.

ОПК-5.2. Применяет язык программирования C++ и методы программирования для разработки алгоритмов и компьютерных программ, пригодных для практического применения.

4.3. Результаты обучения

ОПК-2.2.1. Знает язык программирования C++ и методы программирования, методы решения задач, приёмы доказательства утверждений, методы интегральных преобразований, применяемые для решения профессиональных задач.

ОПК-2.2.2. Умеет выбирать и использовать необходимые знания по языку программирования C++ и методам программирования для использования и адаптации существующих математических методов и систем программирования при разработке и реализации алгоритмов решения прикладных задач.

ОПК-2.2.3. Владеет языком программирования C++ и методами программирования при использовании и адаптации существующих математических методов и систем программирования для разработки и реализации алгоритмов решения прикладных задач.

ОПК-5.2.1. Знает язык программирования C++ и методы программирования и аргументированно может разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.

ОПК-5.2.2. Умеет программировать на языке программирования C++ и может разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.

ОПК-5.2.3. Владеет базовыми методами программирования и может разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.

5. ПРОГРАММА ДИСЦИПЛИНЫ

Название темы	Краткое содержание темы (вопросы темы)
Раздел 1. Программирование простейших алгоритмов	
1. Введение в программирование	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.
2. Алгоритмический язык С++ и его простейшие конструкции	2.1. Алгоритмический язык С++. Символы и основные лексемы языка. Структура программы. Типы данных, их представления в памяти. 2.2. Константы, переменные, массивы базовых типов, указатели. 2.3. Выражения, операнды, операции, их содержание и особенности выполнения.
3. Программирование линейных и разветвляющихся вычислительных процессов	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.
4. Программирование простых циклических вычислительных процессов	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.
Раздел 2. Программирование вложенных циклов и построение простейших программ модульной структуры	
5. Вложенные циклы	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.
6. Функции, построение простейших многомодульных программ	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.
Раздел 3. Программирование вычислительных процессов с обработкой одномерных и двумерных массивов	
7. Проектирование и составление многомодульных программ для	7.1. Связь между формальными и фактическими параметрами в общем случае. 7.2. Методика проектирования и составления многомодульных

формирования и обработки элементов одномерных массивов	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).
8. Проектирование и составление многомодульных программ для формирования и обработки элементов многомерных массивов	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).
9. Проектирование и составление многомодульных программ реализации некоторых численных методов	Составление многомодульных программ для организации некоторых вычислительных процессов (вычисление определенных интегралов и решение трансцендентных уравнений)
Раздел 4. Программирование сложных вычислительных процессов	
10. Программирование сложных вычислительных процессов	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
11. Введение в объектно-ориентированное программирование	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.

6. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

6.1. Форма обучения – очная, курс – 1, семестр – 1

Наименования разделов и тем	Количество часов				
	Лекц.	Лабор.	Практ.	СРС+К	Всего
Раздел 1. Программирование простейших алгоритмов	18	34		61	113
1. Введение в программирование	6	2		8	16
2. Алгоритмический язык C++ и его	6	4		15	25

простейшие конструкции					
3. Программирование линейных и разветвляющихся процессов	2	8		12	22
4. Программирование простых циклических вычислительных процессов	4	20		26	50
Раздел 2. Программирование вложенных циклов и построение простейших программ модульной структуры	16	34		53	103
5. Вложенные циклы	8	18		32	58
6. Функции, построение простейших многомодульных программ	8	16		21	45
ИТОГО ЗА СЕМЕСТР	34	68		114	216

6.2. Форма обучения – очная, курс – 1, семестр – 2

Наименования разделов и тем	Количество часов				
	Лекц.	Лабор.	Практ.	СРС+К	Всего
Раздел 3. Программирование вычислительных процессов с обработкой одномерных и двумерных массивов	16	44		50	110
7. Проектирование и составление многомодульных программ для формирования и обработки элементов одномерных массивов *	6	16		15	37
8. Проектирование и составление многомодульных программ для формирования и обработки элементов двумерных массивов*	6	16		20	42
9. Проектирование и составление многомодульных программ реализации некоторых численных методов*	4	12		15	31
Раздел 4. Программирование сложных вычислительных процессов	14	16		40	70
10. Программирование сложных вычислительных процессов*	6	16		25	47
11. Введение в объектно-ориентированное программирование*	8	-		15	23
ИТОГО ЗА СЕМЕСТР	30	60		90	180
ИТОГО ПО КОМПОНЕНТУ ОПОП	64	128		235	396

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

7.1. Контрольные вопросы

Раздел 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 некуррируемыми?
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?

Раздел 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 $\|c_{ij}\| = \|a_{ik}\| \|b_{kj}\|$, 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?

Раздел 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?

Раздел 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 `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?

Организация самостоятельной работы обучающихся

Самостоятельная работа имеет особенное значение для креативного (творческого) усвоения основных понятий и категорий основы научной работы обучающихся. Самостоятельная работа обучающегося является важной формой учебного процесса, которая позволяет приобрести, а также закрепить новые знания, навыки и умения, сформировать личные убеждения, использовать полученные знания и умения в практической деятельности. Она осуществляется на протяжении всего процесса обучения и имеет следующие стадии:

1. Первичное ознакомление с теоретическим материалом и составление конспекта;
2. Изучение и усвоение теоретического материала;
3. Самостоятельная проработка литературных источников и обобщение изученного материала;

4. Подготовка к практическим (лабораторным) занятиям;
5. Выполнение практических заданий;
6. Индивидуальная работа по заданию преподавателя.

Контрольными формами самостоятельной работы по дисциплине могут быть следующие: работа с литературными первоисточниками по темам дисциплины; выполнение практических заданий, подготовка докладов, тезисов, научных статей.

7.2. Темы письменных работ (типы задач)

Контрольная работа по проверке теоретических знаний – по всем темам, с использованием указанных выше контрольных вопросов.

Образец задания модульного контроля

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

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

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

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

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

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.

7.3. Образец содержания экзаменационного билета

Экзаменационный 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.

В случае ведения учебного процесса с использованием электронного обучения и дистанционных образовательных технологий, содержание билета может отличаться от приведенного.

8. РАСПРЕДЕЛЕНИЕ БАЛЛОВ, КОТОРЫЕ ПОЛУЧАЮТ ОБУЧАЮЩИЕСЯ

Общая оценка знаний обучающихся по дисциплине проводится по 100-балльной шкале исходя из максимума, приведенного в таблице ниже. Организационно-учебная работа в аудитории оценивается на основе таких критериев как посещаемость занятий, своевременное и качественное выполнение домашних заданий, активность во время проведения лекционных и лабораторных занятий (участие в обсуждении текущего и пройденного материала, решение задач и т.п.).

Если суммарное количество полученных студентом в течение каждого семестра баллов достаточно для положительной оценки и, если эта оценка его устраивает, то он освобождается от экзамена с сохранением ему этих баллов. В остальных случаях студенту сохраняется 20% этих баллов, а остальные баллы он получает на экзамене, где он может получить до 80 баллов.

8.1. Семестр 1

Номера разделов	Виды работ	Максимальное количество баллов
1-2	Организационно-учебная работа студента в аудитории	5
	Самостоятельная работа и лабораторные работы	70
	Модульная контрольная работа	25
ИТОГО		100

Экзамен	80 и 20% от успехов в семестре
Общий итог за семестр	100

8.2.Семестр 2

Номера разделов	Виды работ	Максимальное количество баллов
3-4	Организационно-учебная работа студента в аудитории	5
	Самостоятельная работа и лабораторные работы	70
	Модульная контрольная работа	25
ИТОГО		100
Экзамен		80 и 20% от успехов в семестре
Общий итог за семестр		100

Соответствие баллов оценке

Количество баллов из 100	ECTS	Оценка по пятибалльной шкале
		Экзамен
90-100	A	отлично
80-89	B	хорошо
75-79	C	
70-74	D	
60-69	E	удовлетворительно
35-59	FX	
0-34	F	

9. ОБЕСПЕЧЕНИЕ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ И ИНВАЛИДОВ

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

- 1) для слепых и слабовидящих:
 - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
 - для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств;
 - письменные задания оформляются увеличенным шрифтом.
- 2) для глухих и слабослышащих:
 - лекции оформляются в виде электронного документа;
 - письменные задания выполняются на компьютере в письменной форме;
 - экзамен проводится в письменной форме на компьютере; возможно проведение в форме тестирования.
- 3) для лиц с нарушениями опорно-двигательного аппарата:
 - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;

- письменные задания выполняются на компьютере;
- экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

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

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

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

- 1) для слепых и слабовидящих:
 - в печатной форме увеличенным шрифтом;
 - в форме электронного документа;
- 2) для глухих и слабослышащих:
 - в печатной форме;
 - в форме электронного документа.
- 3) для обучающихся с нарушениями опорно-двигательного аппарата:
 - в печатной форме;
 - в форме электронного документа.

10. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОГО ПРОЦЕССА

Учебные занятия проводятся в главном корпусе ДонГУ (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.».

11. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

11.1. 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.

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.
3. Шилдт Г. Самоучитель С++ : [Для программистов и опыт. пользователей] / Герберт Шилдт ; [Пер. с англ. Алексей Жданов]. – 3-е изд. – СПб. и др. : BHV-Санкт-Петербург, 2001. – 683 с.
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.
- 11.2. Дополнительная литература
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.
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.
7. Подбельский В.В. Язык Си++. – М.: Финансы и статистика, 1996. – 560 с.
8. 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.
9. Сван Т. Освоение Borland С++ 4.5. Практический курс. К.: Диалектика, 1996. – 544 с.
10. 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.
11. Абрамов С. А. Задачи по программированию / С. А. Абрамов, Г. Г. Гнездилова, Е. Н. Капустина, М. И. Седюн. – М. : Наука, 1988. – 224 с.

12. ИНФОРМАЦИОННЫЕ РЕСУРСЫ

1. Научная электронная библиотека elibrary.ru : информ.-аналит. портал / ООО Научная электронная библиотека. – Москва : ООО Науч. электрон. б-ка, сор. 2000–2022. – URL: <https://elibrary.ru> (дата обращения: 01.01.2023). – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.
2. Электронный каталог Научной библиотеки Донецкого государственного университета. – Донецк : НБ ДонГУ, 1999– . – URL: <http://catalog.donnu.education> (дата обращения: 01.01.2023). – Текст : электронный;
3. Техническая библиотека URL: <http://techlibrary.ru/> (дата обращения: 31.03.2023). – Режим доступа: свободный. – Текст : электронный;
4. Научные журналы ФГБОУ ВО «ДонГУ» URL: <http://donnu.ru/science/journals> (дата обращения: 31.03.2023). – Режим доступа: свободный. – Текст : электронный.
5. Word: справка и обучение URL: <https://support.office.com/ru-ru/word> (дата обращения: 31.03.2023) . – Режим доступа: свободный. – Текст : электронный.
6. Справка и обучение по Excel URL: <https://support.office.com/ru-ru/word> (дата обращения: 31.03.2023). – Режим доступа: свободный. – Текст : электронный.
7. Статьи по ИТ и алгоритмам URL: <https://proglib.io/> (дата обращения: 31.03.2023) . – Режим доступа: свободный. – Текст : электронный.
8. Интегрированная среда разработки Visual Studio: документация URL: <https://learn.microsoft.com/ru-ru/visualstudio/windows/?view=vs-2022&preserve-view=true> (дата обращения: 31.03.2023) . – Режим доступа: свободный. – Текст : электронный.

9. Куликов А.С. Лекции Видеокурса «Алгоритмы и структуры данных». URL: <https://www.lektorium.tv/course/22823> (дата обращения: 31.03.2023) . – Режим доступа: свободный. – Текст : электронный.

10. Подборка алгоритмов на разную тематику и их реализация на C++. URL: <https://e-maxx.ru/algo/> (дата обращения: 31.03.2023) . – Режим доступа: свободный. – Текст : электронный.

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 для свободного программного обеспечения: Антивирус Касперского, Libre Office, Adobe Acrobat Reader, xPDF, Paint.NET.