

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Институт информационных технологий, математики и механики

УТВЕРЖДЕНО

решением президиума Ученого совета ННГУ

протокол № 1 от 16.01.2024 г.

Рабочая программа дисциплины

Языки и методы программирования

Уровень высшего образования

Бакалавриат

Направление подготовки / специальность

01.03.02 - Прикладная математика и информатика

Направленность образовательной программы

Прикладная математика и информатика (общий профиль)

Форма обучения

очная

г. Нижний Новгород

2024 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.О.05 Языки и методы программирования относится к обязательной части образовательной программы.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства	
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	Для текущего контроля успеваемости	Для промежуточной аттестации
ОПК-2: Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	ОПК-2.1: Знает математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач ОПК-2.2: Умеет осуществлять выбор и адаптацию математических методов и программного обеспечения для разработки и реализации алгоритмов решения задач в области профессиональной деятельности ОПК-2.3: Имеет практический опыт применения математических методов и систем программирования для разработки и реализации алгоритмов решения прикладных задач	ОПК-2.1: Знает общие принципы использования вычислительной техники для решения прикладных задач; основные идеи, лежащие в основе современных языков программирования и основные составляющие современного языка программирования на примере C и C++; технологии разработки программного обеспечения (структурное и модульное программирование – подробно, объектно-ориентированное программирование – на уровне общих идей) и способы их выражения в языках программирования (C, ООП – в C++); методы конструирования новых типов данных (C, C++); методы и приемы работы с динамической памятью (C, C++); методы работы с внешней памятью (C, C++). ОПК-2.2: Умеет разрабатывать и отлаживать программы средней сложности с использованием языков программирования C, C++ и технологий структурного, модульного и объектно-	Задачи Практическое задание	Зачёт: Задания Тест Экзамен: Контрольные вопросы

		<p>ориентированного программирования.</p> <p>ОПК-2.3: Владеет современными интегрированными средами разработки программ; навыками реализации, тестирования и отладки программных систем средней сложности (на базовом уровне).</p>		
ОПК-5: Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	<p>ОПК-5.1: Знает основы алгоритмизации и основы программирования, один или несколько языков программирования</p> <p>ОПК-5.2: Умеет разрабатывать алгоритмы и компьютерные программы для практического применения</p> <p>ОПК-5.3: Имеет практический опыт разработки алгоритмов и компьютерных программ для практического применения</p>	<p>ОПК-5.1: Знает элементы анализа и разработки алгоритмов, алгоритмы поиска и сортировки данных.</p> <p>ОПК-5.2: Умеет разрабатывать алгоритмы и компьютерные программы с использованием языков программирования C, C++, пригодные для практического применения.</p> <p>ОПК-5.3: Владеет практическими навыками разработки алгоритмов и компьютерных программ с использованием языков программирования C, C++, пригодных для практического применения.</p>	Задачи Практическое задание	<p>Зачёт: Задания Тест</p> <p>Экзамен: Контрольные вопросы</p>

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очная
Общая трудоемкость, з.е.	12
Часов по учебному плану	432
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	128
- занятия семинарского типа (практические занятия / лабораторные работы)	128
- КСР	3

самостоятельная работа	137
Промежуточная аттестация	36 Экзамен, Зачёт

3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

Наименование разделов и тем дисциплины	Всего (часы)	в том числе			
		Контактная работа (работа во взаимодействии с преподавателем), часы из них			Самостоятельная работа обучающегося, часы
		Занятия лекционного типа	Занятия семинарского типа (практические занятия/лабораторные работы), часы	Всего	
	0 Ф 0	0 Ф 0	0 Ф 0	0 Ф 0	0 Ф 0
1.1. Введение в предмет.	2	2	0	2	0
1.2. Решение задач с использованием вычислительной техники.	5	2	0	2	3
1.3. Современная система разработки программного обеспечения.	8	4	0	4	4
1.4. Среда исполнения программ. Программа в среде MicrosoftWindows.	8	4	0	4	4
1.5. Основные понятия языков программирования.	12	4	4	8	4
1.6. Типы данных, способы и механизмы управления данными.	22	10	8	18	4
1.7. Программа на языке C. Методы и основные этапы трансляции.	10	2	4	6	4
1.8. Структурное программирование и операторы языка C.	26	10	12	22	4
1.9. Конструирование новых типов данных.	26	10	12	22	4
1.10. Модульное программирование.	19	6	8	14	5
1.11. Элементы анализа и разработки алгоритмов.	15	2	8	10	5
1.12. Методы работы с внешней памятью. Файлы.	13	4	4	8	5
1.13. Динамическое управление памятью.	13	4	4	8	5
2.1. Введение в предмет. История языка C++, обзор литературы.	10	4	2	6	4
2.2. Элементы языка C++, от C к C++ шаг за шагом.	16	8	4	12	4
2.3. Структурное программирование и операторы языка C++.	10	2	4	6	4
2.4. Массивы и указатели в C++. Работа со строками.	14	6	4	10	4
2.5. Модульное программирование в C++.	14	4	4	8	6
2.6. Структуры и объединения в C++.	8	2	2	4	4
2.7. Работа с файлами в C++.	12	4	4	8	4
2.8. Язык программирования C++. Дополнительные главы.	16	6	4	10	6
2.9. Концепция объектного подхода. Основные идеи объектного подхода на примерах.	12	2	4	6	6
2.10. Понятие класса. Классы в C++.	14	4	4	8	6
2.11. Конструкторы.	12	4	4	8	4
2.12. Деструкторы.	10	2	4	6	4
2.13. Статические поля и методы. Дружественные функции.	12	2	4	6	6
2.14. Перегрузка операций.	14	4	4	8	6
2.15. Наследование и агрегация	13	3	4	7	6

2.16. Виртуальные методы. Абстрактные классы.	13	3	4	7	6
2.17. Введение в шаблоны	14	4	4	8	6
Аттестация	36				
КСР	3			3	
Итого	432	128	128	259	137

Содержание разделов и тем дисциплины

Семестр 1.

1. Введение в предмет. Структура и содержание курса. Литература.
2. Решение задач с использованием вычислительной техники.
3. Современная система разработки программного обеспечения.
4. Среда исполнения программ. Программа в среде Microsoft Windows.
5. Основные понятия языков программирования. Синтаксис, семантика, формальные способы описания языков программирования.
6. Типы данных, способы и механизмы управления данными.
7. Программа на языке C. Методы и основные этапы трансляции.
8. Структурное программирование и операторы языка C.
9. Конструирование новых типов данных.
10. Модульное программирование.
11. Элементы анализа и разработки алгоритмов.
12. Методы работы с внешней памятью. Файлы.
13. Динамическое управление памятью.

Семестр 2.

1. Введение в предмет. История языка C++, обзор литературы.
2. Элементы языка C++, от C к C++ шаг за шагом. Переменные, типы данных, константы, литералы, перечисления, объявления, операции, математические функции, преобразование типов, ввод/вывод.
3. Структурное программирование и операторы языка C++.
4. Массивы и указатели в C++. Работа со строками.
5. Модульное программирование в C++.
6. Структуры и объединения в C++.
7. Работа с файлами в C++.
8. Язык программирования C++. Дополнительные главы (работа с динамической памятью, новые стандарты и возможности языка, основные направления развития, элементы стандартной библиотеки).
9. Концепция объектного подхода. Основные идеи объектного подхода на примерах.
10. Понятие класса. Классы в C++. Поля и методы. Разграничение доступа. Классы и объекты. Константные методы.
11. Конструкторы.
12. Деструкторы.
13. Статические поля и методы. Дружественные функции.
14. Перегрузка операций.
15. Наследование и агрегация
16. Виртуальные методы. Абстрактные классы.
17. Введение в шаблоны

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

Для обеспечения самостоятельной работы обучающихся используются:
Электронные курсы, созданные в системе электронного обучения ННГУ:

Языки и методы программирования, <https://e-learning.unn.ru/enrol/index.php?id=1509>.

Иные учебно-методические материалы:

1) Сысоев А.В. Основы программирования: Учебно-методическое пособие. – Нижний Новгород: Нижегородский госуниверситет, 2024. – 32 с.

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Задачи) для оценки сформированности компетенции ОПК-2:

Задача 1. Дан массив координат точек на плоскости. Вывести индексы точек, наиболее близких друг к другу.

Задача 2. Написать класс "Простая дробь". Реализовать конструкторы, деструктор, перегрузку арифметических операторов. Продемонстрировать корректность работы программы на примере.

5.1.2 Типовые задания (оценочное средство - Задачи) для оценки сформированности компетенции ОПК-5:

Задача 1. Задан массив A, содержащий n вещественных чисел. Проверить, является ли массив упорядоченным по возрастанию.

Задача 2. Задан массив A, содержащий n целых чисел и вещественное число R. Найти индексы двух элементов массива, сумма которых наиболее близка к числу R.

Критерии оценивания (оценочное средство - Задачи)

Оценка	Критерии оценивания
зачтено	Выполнена основная часть задания, возможно с незначительными недочетами
не зачтено	Выполнено менее половины задания, есть существенные недочеты

5.1.3 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ОПК-2:

1. Разработка программы сортировки и поиска данных.
2. Разработка программы для вычисления значения функций с помощью суммирования рядов.
3. Разработка программы для решения систем алгебраических линейных уравнений.

5.1.4 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ОПК-5:

1. Разработка класса «Комплексное число».
2. Разработка класса «Вектор».

Критерии оценивания (оценочное средство - Практическое задание)

Оценка	Критерии оценивания
зачтено	Выполнены все или большая часть этапов решения задачи или задача решена с незначительными недочетами. Программа и результаты работы представлены преподавателю в срок.
не зачтено	Выполнены не все задания или выполнены не в полном объеме (представлено не полное описание этапов выполнения заданий, программа работает некорректно, результаты работы не представлены преподавателю).

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатора достижения компетенций)	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	не зачтено		зачтено				
<u>Знания</u>	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	Уровень знаний ниже минимальных требований. Имели место грубые ошибки	Минимально допустимый уровень знаний. Допущено много негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет.	Уровень знаний в объеме, превышающем программу подготовки.

<u>Умения</u>	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые с недочетами	Продемонстрированы все основные умения. Решены все основные задачи с отдельным и несущественными недочетами, выполнены все задания в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
<u>Навыки</u>	Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки	Имеется минимальный набор навыков для решения стандартных задач с некоторым и недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторым и недочетами	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов	Продемонстрирован творческий подход к решению нестандартных задач

Шкала оценивания при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой
	отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично».
	очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо».
	удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
не зачтено	неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно».
	плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Задания) для оценки сформированности компетенции ОПК-2

1. Написать программу на языке C, выполняющую сортировку целочисленного массива по возрастанию.
2. Разработать класс "Комплексное число". Реализовать конструкторы, деструктор, выполнить перегрузку арифметических операций.

5.3.2 Типовые задания (оценочное средство - Задания) для оценки сформированности компетенции ОПК-5

1. Написать программу на языке C, позволяющую вычислить значение тригонометрической функции в заданной точке через разложение в ряд Тейлора.
2. Разработать класс "Вектор". Реализовать конструкторы, деструктор, выполнить перегрузку арифметических операций.

Критерии оценивания (оценочное средство - Задания)

Оценка	Критерии оценивания
зачтено	Выполнена основная часть задания, возможно с незначительными недочетами
не зачтено	Выполнено менее половины задания, есть существенные недочеты

5.3.3 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ОПК-2

1. Тип вопроса: одиночный выбор

Укажите правильный вариант объявления переменной целого типа в языках C и C++:

- a. int a;
- b. float a;
- c. double a;

1. Тип вопроса: одиночный выбор

Укажите правильный вариант объявления массива с целочисленными элементами в языках C и C++:

- a. int a;
- b. int a[100];
- c. double a[];
- d. float a[];

1. Тип вопроса: одиночный выбор

Укажите, что будет выведено на экран в результате работы программы на языке C (приведен фрагмент функции main()):

```
int a;  
int *pa;  
a = 1;  
pa = &a;  
printf("%d", *pa);
```

- a. 0
- b. 1
- c. -1
- d. Ошибка времени исполнения

5.3.4 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ОПК-5

1. Тип вопроса: одиночный выбор

В чем смысл использования ключевых слов public и private в объявлении классов на языке C++?

- a. Корректное освобождение памяти
- b. Корректное выделение памяти
- c. Скрытие деталей реализации (+)

1. Тип вопроса: одиночный выбор

Для чего предназначен конструктор класса?

- a. Для создания объектов класса (+)
- b. Для удаления объектов класса
- c. Для объединения объектов класса в единое целое
- d. Для организации взаимодействия классов

1. Тип вопроса: одиночный выбор

Для чего предназначен деструктор класса?

- a. Для создания объектов класса
- b. Для удаления объектов класса (+)
- c. Для объединения объектов класса в единое целое
- d. Для организации взаимодействия классов

Критерии оценивания (оценочное средство - Тест)

Оценка	Критерии оценивания
зачтено	Студент работал на практических занятиях, более 80% правильных ответов в тесте.

Оценка	Критерии оценивания
не зачтено	Студент пропустил большую часть практических занятий, менее 80% правильных ответов в тесте.

5.3.5 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ОПК-2

1. Решение задач с использованием вычислительной техники. Основные этапы их назначения.
2. Современная система разработки программного обеспечения. Основные компоненты и их назначение.
3. Среда исполнения программ. Программа в среде Microsoft Windows.
4. Основные понятия языков программирования. Синтаксис, семантика, формальные способы описания языков программирования.
5. Понятие типа данных. Примеры.
6. Концепция структурного программирования.
7. Состав языка С. Операторы, виды операторов, примеры. Блок (составной оператор). Правила объявления переменных.
8. Структура С-программы. Функция main, варианты объявления.
9. Глобальные и локальные переменные. Описание и употребление констант.
10. Данные целых типов. Операции над целочисленными данными. Примеры использования.
11. Данные вещественных типов. Операции над вещественными данными. Примеры использования.
12. Данные символьного типа. Значение и код символа. Операции. Примеры использования.
13. Объявление типов и использование имен типов для описания данных. Примеры.
14. Массивы, объявления. Операции над массивами. Двумерные массивы, объявления, размещение элементов в памяти.
15. Объявление и использование объектов типа "структура". Структуры с битовыми полями.
16. Макросы. Объявление, использование, примеры. Плюсы и минусы.
17. Указатели, объявление, представление в памяти. Функции выделения/освобождения памяти. Адресная арифметика. «Массивная» нотация при работе с указателями.
18. Ввод/вывод в языке С, функции printf, scanf, формат, примеры.
19. Условный оператор. Полный и неполный варианты. Примеры использования.
20. Оператор множественного выбора. Примеры использования.
21. Операторы цикла. Назначение операторов break и continue. Примеры использования.
22. Операторы цикла. Итерационный цикл for. Примеры использования.
23. Операторы цикла. Цикл с предусловием while. Примеры использования.
24. Операторы цикла. Цикл с постусловием do. Примеры использования.
25. Объявление, реализация и использование функций. Локальные и глобальные переменные. Способы передачи параметров.
26. Рекурсивные функции. Примеры.
27. Особенности использования массивов в качестве параметров процедур и функций.
28. Тип указатель на функцию. Примеры использования.
29. Файлы и работа с ними. Функции fopen, fclose, fseek, fscanf, fprintf.
30. Многофайловая программа в языке С. Заголовочные файлы, директивы препроцессора. Условная компиляция.
31. Технологии программирования: структурное, модульное, ООП. Общее описание.

32. Концепции ООП: абстрагирование, инкапсуляция. Поддержка в языке программирования C++. Примеры.
33. Концепции ООП: наследование. Поддержка в языке программирования C++. Примеры.
34. Концепции ООП: полиморфизм. Поддержка в языке программирования C++. Примеры.
35. Создание объектов. Конструкторы. Виды конструкторов, объявление. Реализация конструкторов по умолчанию и инициализации на примере. Примеры использования.
36. Создание объектов. Конструкторы. Виды конструкторов, объявление. Реализация конструкторов копирования и преобразования типа на примере класса Complex. Примеры использования.
37. Удаление объектов. Деструктор. Пример реализации для класса NamedComplex. Реализация метода Add для класса Complex.
38. Перегрузка унарных и бинарных операций. Перегрузка префиксного и постфиксного инкремента. Примеры использования.
39. Перегрузка унарных и бинарных операций. Перегрузка операций + и += для класса Complex. Примеры использования.
40. Перегрузка унарных и бинарных операций. Перегрузка оператора присваивания на примере класса Vector. Отличия оператора присваивания от конструктора копирования. Примеры использования.
41. Спецификаторы доступа в объявлении класса. Примеры. Дружественные функции.
42. Полиморфизм на примере иерархии классов геометрических фигур. Абстрактные классы. Виртуальный деструктор.
43. Полиморфизм на примере иерархии классов геометрических фигур. Таблица виртуальных функций. Преимущества использования полиморфизма.
44. Шаблоны классов. Описание. Отличия реализации шаблонного класса от обычного. Примеры.

5.3.6 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ОПК-5

1. Элементы анализа и разработки алгоритмов. Алгоритмы поиска данных в массивах.
2. Элементы анализа и разработки алгоритмов. Алгоритмы сортировки данных.
3. Элементы анализа и разработки алгоритмов. Алгоритмы сортировки данных. Быстрая сортировка.00
4. Разработка класса Vector. Объявление (конструкторы, деструктор, перегрузка операций +, -, +=, -=, ++, --, [], =). Реализация конструкторов, деструктора, оператора присваивания.
5. Разработка класса Vector. Объявление (конструкторы, деструктор, перегрузка операций +, -, +=, -=, ++, --, [], =). Реализация операций +, +=, ++, [].

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
превосходно	Студент дал развернутый ответ на все вопросы и при этом продемонстрировал знание дополнительного материала.
отлично	Студент дал развернутый ответ на все вопросы.
очень хорошо	Студент дал ответ на все вопросы, возможно с незначительными недочетами.
хорошо	Студент ответил на большую часть вопросов с незначительными недочетами.
удовлетворительно	Студент ответил на большую часть вопросов с существенными недочетами.

Оценка	Критерии оценивания
неудовлетворительно	При ответе студент допускает грубые ошибки в основном материале и решении стандартных задач.
плохо	Отсутствие знаний материала, отсутствует способность решения стандартных задач.

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Подбельский В.В. Курс программирования на языке Си : учебник / Подбельский В.В.; Фомин С.С. - Москва : ДМК-пресс, 2018. - 384 с. - ISBN 978-5-97060-229-4., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=772960&idb=0>.
2. Язык программирования C++ для профессионалов / Страуструп Б. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=663690&idb=0>.
3. Язык программирования C / Керниган Б.В., Ричи Д.М. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=663682&idb=0>.
4. Трофимов Валерий Владимирович. Алгоритмизация и программирование : учебник для вузов / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. - 4-е изд. - Москва : Юрайт, 2024. - 118 с. - (Высшее образование). - URL: <https://urait.ru/bcode/538039> (дата обращения: 15.08.2024). - ISBN 978-5-534-17497-7 : 539.00. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=902856&idb=0>.

Дополнительная литература:

1. Язык программирования C++ / Фридман А.Л. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=663697&idb=0>.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

1. Баженова И., Сухомлин В. Курс «Введение в программирование». <http://www.intuit.ru/studies/courses/27/27/info>
2. Страуструп Б. Курс «Язык программирования C++ для профессионалов». <http://www.intuit.ru/studies/courses/98/98/info>
3. Калинина Н., Костюкова Н. Курс «Основы программирования на языке C». <http://www.intuit.ru/studies/courses/43/43/info>
4. Фридман А. Курс «Язык программирования C++». <http://www.intuit.ru/studies/courses/17/17/info>

Используется следующее лицензионное программное обеспечение:

1. Операционные системы семейства Microsoft Windows, – лицензия по подписке Microsoft Imagine;
2. Среды разработки семейства Microsoft Visual Studio, – лицензия по подписке Microsoft Imagine;

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 01.03.02 - Прикладная математика и информатика.

Автор(ы): Мееров Иосиф Борисович, кандидат технических наук, доцент
Сысоев Александр Владимирович, кандидат технических наук.

Заведующий кафедрой: Мееров Иосиф Борисович, кандидат технических наук.

Программа одобрена на заседании методической комиссии от 13.12.2023, протокол № 3.