

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
(факультет / институт / филиал)

УТВЕРЖДЕНО
решением Ученого совета ННГУ
от 30.11.2022 г. протокол № 13

Рабочая программа дисциплины

Языки и методы программирования

(наименование дисциплины (модуля))

Уровень высшего образования

Бакалавриат

(бакалавриат / магистратура / специалитет)

Направление подготовки / специальность

01.03.02 Прикладная математика и информатика

(указывается код и наименование направления подготовки / специальности)

Направленность образовательной программы

Математическое моделирование и вычислительная математика

(указывается профиль / магистерская программа / специализация)

Форма обучения

очная

(очная / очно-заочная / заочная)

Нижегород

2023 год

1. Место дисциплины в структуре ООП

Дисциплина относится к обязательной части.

Б1.О.05 « Языки и методы программирования».

№ варианта	Место дисциплины в учебном плане образовательной программы	Стандартный текст для автоматического заполнения в конструкторе РПД
1	Блок 1. Дисциплины (модули) Обязательная часть	Дисциплина Б1.О.05 « Языки и методы программирования» относится к обязательной части ООП направления подготовки 01.03.02 <i>Прикладная математика и информатика</i> ».

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	
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	Знать 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++).	Собеседование тест
	ОПК-2.2. Умеет осуществлять выбор и адаптацию математических	Уметь разрабатывать и отлаживать программы средней сложности с использованием языков программирования C, C++	Задачи Задания

	<i>методов и программного обеспечения для разработки и реализации алгоритмов решения задач в области профессиональной деятельности</i>	<i>и технологий структурного, модульного и объектно-ориентированного программирования.</i>	
	<i>ОПК-2.3. Имеет практический опыт применения математических методов и систем программирования для разработки и реализации алгоритмов решения прикладных задач</i>	<i>Владеть современными интегрированными средами разработки программ; навыками реализации, тестирования и отладки программных систем средней сложности (на базовом уровне).</i>	<i>Задачи Задания</i>
<i>ОПК-5. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения</i>	<i>ОПК-5.1. Знает основы алгоритмизации и основы программирования, один или несколько языков программирования</i>	<i>Знать элементы анализа и разработки алгоритмов, алгоритмы поиска и сортировки данных</i>	<i>Собеседование тест</i>
	<i>ОПК-5.2. Умеет разрабатывать алгоритмы и компьютерные программы для практического применения</i>	<i>Уметь разрабатывать алгоритмы и компьютерные программы с использованием языков программирования C, C++, пригодные для практического применения</i>	<i>Задачи Задания</i>
	<i>ОПК-5.3. Имеет практический опыт разработки алгоритмов и компьютерных программ для практического применения</i>	<i>Владеть практическими навыками разработки алгоритмов и компьютерных программ с использованием языков программирования C, C++, пригодных для практического применения</i>	<i>Задачи Задания</i>

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

Семестр 1

	Очная форма обучения
Общая трудоемкость	5 ЗЕТ
Часов по учебному плану	180
в том числе	
аудиторные занятия (контактная работа):	129
- занятия лекционного типа	64
- занятия семинарского типа	32
- занятия лабораторного типа	32
- текущий контроль (КСР)	1
самостоятельная работа	51
Промежуточная аттестация – зачет	

Семестр 2

	Очная форма обучения
Общая трудоемкость	7 ЗЕТ
Часов по учебному плану	252
в том числе	
аудиторные занятия (контактная работа):	130
- занятия лекционного типа	64
- занятия семинарского типа	32
- занятия лабораторного типа	32
- текущий контроль (КСР)	2
самостоятельная работа	86
Промежуточная аттестация – экзамен	36

3.2. Содержание дисциплины

Наименование и краткое содержание разделов и тем дисциплины	Всего (часы)	В том числе				Самостоятельная работа обучающегося, часы
		Контактная работа (работа во взаимодействии с преподавателем), часы. Из них				
		Занятия лекционного типа	Занятия семинарского типа	Занятия лабораторного типа	Всего	

	Очная	Очная	Очная	Очная	Очная	Очная
Семестр 1						
1. Введение в предмет. Структура и содержание курса. Литература.	2	2			2	
2. Решение задач с использованием вычислительной техники.	5	2			2	3
3. Современная система разработки программного обеспечения.	8	4			4	4
4. Среда исполнения программ. Программа в среде Microsoft Windows.	8	4			4	4
5. Основные понятия языков программирования. Синтаксис, семантика, формальные способы описания языков программирования.	12	4	2	2	8	4
6. Типы данных, способы и механизмы управления данными.	22	10	4	4	18	4
7. Программа на языке C. Методы и основные этапы трансляции.	10	2	2	2	6	4
8. Структурное программирование и операторы языка C.	26	10	6	6	22	4
9. Конструирование новых типов данных.	26	10	6	6	22	4
10. Модульное программирование.	19	6	4	4	14	5
11. Элементы анализа и разработки алгоритмов.	15	2	4	4	10	5
12. Методы работы с внешней памятью. Файлы.	13	4	2	2	8	5
13. Динамическое управление памятью.	13	4	2	2	8	5
Текущий контроль (КСР)	1				1	
Промежуточная аттестация – зачет						
Итого	180		32	32	129	51
Семестр 2						
1. Введение в предмет. История языка C++, обзор литературы.	10	4	1	1	6	4
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.	16	8	2	2	12	4
3. Структурное программирование и операторы языка C++.	10	2	2	2	6	4
4. Массивы и указатели в C++. Работа со строками.	14	6	2	2	10	4
5. Модульное программирование в C++.	14	4	2	2	8	6

6.Структуры и объединения в C++.	8	2	1	1	4	4
7.Работа с файлами в C++.	12	4	2	2	8	4
8.Язык программирования C++. Дополнительные главы (работа с динамической памятью, новые стандарты и возможности языка, основные направления развития, элементы стандартной библиотеки).	16	6	2	2	10	6
9.Концепция объектного подхода. Основные идеи объектного подхода на примерах.	12	2	2	2	6	6
10.Понятие класса. Классы в C++. Поля и методы. Разграничение доступа. Классы и объекты. Константные методы.	14	4	2	2	8	6
11.Конструкторы.	12	4	2	2	8	4
12.Деструкторы.	10	2	2	2	6	4
13.Статические поля и методы. Дружественные функции.	12	2	2	2	6	6
14.Перегрузка операций.	14	4	2	2	8	6
15.Наследование и агрегация	13	3	2	2	7	6
16.Виртуальные методы. Абстрактные классы.	13	3	2	2	7	6
17.Введение в шаблоны	14	4	2	2	8	6
Текущий контроль (КСР)	2				2	
Промежуточная аттестация – экзамен	36					
Итого	252	64	32	32	130	86

Практические занятия (семинарские занятия/лабораторные занятия) организуются, в том числе в форме практической подготовки, которая предусматривает участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

На проведение практических занятий (семинарских занятий/лабораторных занятий) в форме практической подготовки отводится 64/64 часа.

Практическая подготовка направлена на формирование и развитие:

- практических навыков в соответствии с профилем ОП: навыками реализации, тестирования и отладки программных систем средней сложности4 навыками разработки алгоритмов и компьютерных программ с использованием языков программирования С, С++, пригодных для практического применения
- компетенций – ОПК-2; ОПК-5.

Текущий контроль успеваемости реализуется в рамках занятий семинарского типа, лабораторного типа.

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа студентов в ходе всего учебного года предполагает выполнение ряда практических работ (проектов). При этом в каждой практической работе студенты проходят весь путь, начиная от постановки учебной задачи до сдачи преподавателю работающей программы с краткой документацией. Успешная сдача практических работ является основным критерием при постановке зачета по дисциплине.

а) Виды самостоятельной работы студентов

Самостоятельная работа студентов подразделяется на следующие категории:

- Изучение учебной литературы (см. перечень образовательных материалов).
- Решение учебных задач (основы алгоритмизации, закрепление знаний синтаксиса и семантики языков программирования, развитие навыков практического программирования).
- Выполнение практических работ на следующие темы (по усмотрению преподавателя):
 - Практическая работа 1. Алгоритмы сортировки и поиска данных.
 - Практическая работа 2. Суммирование рядов.
 - Практическая работа 3. Решение систем алгебраических линейных уравнений.
 - Практическая работа 4. Разработка класса «Комплексное число».
 - Практическая работа 5. Разработка класса «Вектор».
- Студенты, демонстрирующие хорошую успеваемость и высокий уровень мотивации, могут принять участие в конкурсе программных проектов, ежегодно проводимом кафедрой Математического обеспечения и суперкомпьютерных технологий института ИТММ.

б) Образовательные материалы для самостоятельной работы студентов

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

Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведены в п. 5.2.

5. Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю), включающий:

5.1. Описание шкал оценивания результатов обучения по дисциплине

Уровень сформированности компетенций (индикатора достижения компетенций)	Шкала оценивания сформированности 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, без ошибок.	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.	Продemonстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме.	Продemonстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продemonстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продemonстрированы все основные умения, решены все основные задачи с отдельными несущественными недочетами, выполнены все задания в полном объеме.	Продemonстрированы все основные умения, решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
<u>Навыки</u>	Отсутствие владения материалом. 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 некоторыми недочетами.	Продemonстрированы базовые навыки при решении стандартных задач с некоторыми недочетами	Продemonстрированы базовые навыки при решении стандартных задач без ошибок и недочетов.	Продemonстрированы навыки при решении нестандартных задач без ошибок и недочетов.	Продemonстрирован творческий подход к решению нестандартных задач.

Шкала оценки при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	Превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно»
	Отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично», при этом хотя бы одна компетенция сформирована на уровне «отлично»
	Очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо», при этом хотя бы одна компетенция сформирована на уровне «очень хорошо»
	Хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо», при этом хотя бы одна компетенция сформирована на уровне «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», ни одна из компетенций не сформирована на уровне «плохо»
	Плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.2 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения

5.2.1 Контрольные вопросы Семестр 1

<i>вопросы</i>	<i>Код формируемой компетенции</i>
1. Решение задач с использованием вычислительной техники. Основные этапы их назначение.	ОПК-2
2. Современная система разработки программного обеспечения. Основные компоненты и их назначение.	ОПК-2
3. Среда исполнения программ. Программа в среде Microsoft Windows.	ОПК-2
4. Основные понятия языков программирования. Синтаксис, семантика, формальные способы описания языков программирования.	ОПК-2
5. Основные элементы языка программирования на примере Object Pascal.	ОПК-2
6. Понятие типа данных. Примеры.	ОПК-2
7. Работа с целыми числами в языке Object Pascal. Примеры.	ОПК-2
8. Работа с целыми числами с плавающей запятой в языке Object Pascal. Примеры.	ОПК-2
9. Работа с символьной и строковой информацией в языке Object Pascal. Примеры.	ОПК-2
10. Работа с логической информацией в языке Object Pascal. Примеры.	ОПК-2
11. Программа на языке Object Pascal. Методы и основные этапы трансляции.	ОПК-2
12. Концепция структурного программирования. Оператор присваивания. Понятие LValue.	ОПК-2

13. Структурное программирование и операторы языка Object Pascal. Ветвления. Примеры.	ОПК-2
14. Структурное программирование и операторы языка Object Pascal. Циклы. Примеры.	ОПК-2
15. Конструирование новых типов данных. Диапазон. Перечислимый тип. Примеры.	ОПК-2
16. Конструирование новых типов данных. Массивы. Примеры.	ОПК-2
17. Конструирование новых типов данных. Записи. Примеры.	ОПК-2
18. Модульное программирование. Основные идеи и принципы. Поддержка в языках.	ОПК-2
19. Модульное программирование. Функции. Передача параметров.	ОПК-2
20. Модульное программирование. Модули. Многомодульные программы.	ОПК-2
21. Элементы анализа и разработки алгоритмов. Алгоритмы поиска данных в массивах.	ОПК-5
22. Элементы анализа и разработки алгоритмов. Алгоритмы сортировки данных.	ОПК-5
23. Элементы анализа и разработки алгоритмов. Алгоритмы сортировки данных. Быстрая сортировка.	ОПК-5
24. Методы работы с внешней памятью. Файлы.	ОПК-2
25. Динамическое управление памятью.	ОПК-2

Семестр 2

<i>вопросы</i>	<i>Код формируемой компетенции</i>
1. Состав языка С. Операторы, виды операторов, примеры. Блок (составной оператор). Правила объявления переменных.	ОПК-2
2. Структура С-программы. Функция main, варианты объявления.	ОПК-2
3. Глобальные и локальные переменные. Описание и употребление констант.	ОПК-2
4. Данные целых типов. Операции над целочисленными данными. Примеры использования.	ОПК-2
5. Данные вещественных типов. Операции над вещественными данными. Примеры использования.	ОПК-2
6. Данные символьного типа. Значение и код символа. Операции. Примеры использования.	ОПК-2
7. Объявление типов и использование имен типов для описания данных. Примеры.	ОПК-2
8. Массивы, объявления. Операции над массивами. Двумерные массивы, объявления, размещение элементов в памяти.	ОПК-2
9. Объявление и использование объектов типа "структура". Структуры с битовыми полями.	ОПК-2
10. Макросы. Объявление, использование, примеры. Плюсы и минусы.	ОПК-2
11. Указатели, объявление, представление в памяти. Функции выделения/освобождения памяти. Адресная арифметика. «Массивная» нотация при работе с указателями.	ОПК-2
12. Ввод/вывод в языке С, функции printf, scanf, формат, примеры.	ОПК-2
13. Условный оператор. Полный и неполный варианты. Примеры использования.	ОПК-2
14. Оператор множественного выбора. Примеры использования.	ОПК-2

15. Операторы цикла. Назначение операторов break и continue. Примеры использования.	ОПК-2
16. Операторы цикла. Итерационный цикл for. Примеры использования.	ОПК-2
17. Операторы цикла. Цикл с предусловием while. Примеры использования.	ОПК-2
18. Операторы цикла. Цикл с постусловием do. Примеры использования.	ОПК-2
19. Объявление, реализация и использование функций. Локальные и глобальные переменные. Способы передачи параметров.	ОПК-2
20. Рекурсивные функции. Примеры.	ОПК-2
21. Особенности использования массивов в качестве параметров процедур и функций.	ОПК-2
22. Тип указатель на функцию. Примеры использования.	ОПК-2
23. Файлы и работа с ними. Функции fopen, fclose, fseek, fscanf, fprintf.	ОПК-2
24. Многофайловая программа в языке C. Заголовочные файлы, директивы препроцессора. Условная компиляция.	ОПК-2
25. Технологии программирования: структурное, модульное, ООП. Общее описание.	ОПК-2
26. Концепции ООП: абстрагирование, инкапсуляция. Поддержка в языке программирования C++. Примеры.	ОПК-2
27. Концепции ООП: наследование. Поддержка в языке программирования C++. Примеры.	ОПК-2
28. Концепции ООП: полиморфизм. Поддержка в языке программирования C++. Примеры.	ОПК-2
29. Создание объектов. Конструкторы. Виды конструкторов, объявление. Реализация конструкторов по умолчанию и инициализации на примере. Примеры использования.	ОПК-2
30. Создание объектов. Конструкторы. Виды конструкторов, объявление. Реализация конструкторов копирования и преобразования типа на примере класса Complex. Примеры использования.	ОПК-2
31. Удаление объектов. Деструктор. Пример реализации для класса NamedComplex. Реализация метода Add для класса Complex.	ОПК-5
32. Перегрузка унарных и бинарных операций. Перегрузка префиксного и постфиксного инкремента. Примеры использования.	ОПК-2
33. Перегрузка унарных и бинарных операций. Перегрузка операций + и += для класса Complex. Примеры использования.	ОПК-2
34. Разработка класса Vector. Объявление (конструкторы, деструктор, перегрузка операций +, -, +=, -=, ++, --, [], =). Реализация конструкторов, деструктора, оператора присваивания.	ОПК-5
35. Разработка класса Vector. Объявление (конструкторы, деструктор, перегрузка операций +, -, +=, -=, ++, --, [], =). Реализация операций +, +=, ++, [].	ОПК-5
36. Перегрузка унарных и бинарных операций. Перегрузка оператора присваивания на примере класса Vector. Отличия оператора присваивания от конструктора копирования. Примеры использования.	ОПК-2
37. Спецификаторы доступа в объявлении класса. Примеры. Дружественные функции.	ОПК-2
38. Полиморфизм на примере иерархии классов геометрических фигур. Абстрактные классы. Виртуальный деструктор.	ОПК-2
39. Полиморфизм на примере иерархии классов геометрических фигур. Таблица виртуальных функций. Преимущества использования полиморфизма.	ОПК-2
40. Шаблоны классов. Описание. Отличия реализации шаблонного класса от обычного. Примеры.	ОПК-2

5.2.2. Типовые тестовые задания 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;`

2. 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[];`

3. 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

5.2.3. Типовые задания для оценки сформированности компетенции ОПК-2

1. Реализовать функцию на языке C, которая решает следующую задачу: дан массив целых чисел. Сформировать из него другой массив, содержащий элементы первого ровно в одном экземпляре (массив без повторений).
2. Реализовать функцию на языке C, которая решает следующую задачу: дана строка. Вернуть данную строку с обратным порядком символов. Вспомогательных строк не создавать.

5.2.4. Типовые 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.

6. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

- 1) Баженова И., Сухомлин В. Курс «Введение в программирование».
<http://www.intuit.ru/studies/courses/27/27/info>
- 2) Страуструп Б. Курс «Язык программирования С++ для профессионалов».
<http://www.intuit.ru/studies/courses/98/98/info>

б) дополнительная литература:

- 3) Калинина Н., Костюкова Н. Курс «Основы программирования на языке С».
<http://www.intuit.ru/studies/courses/43/43/info>
- 4) Фридман А. Курс «Язык программирования С++».
<http://www.intuit.ru/studies/courses/17/17/info>

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

- 1) Microsoft Windows.
- 2) Microsoft Visual Studio.

7. Материально-техническое обеспечение дисциплины

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой, оснащенные оборудованием и техническими средствами обучения.

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

Используется следующее лицензионное программное обеспечение:

1. Операционные системы семейства MicrosoftWindows, – лицензия по подписке MicrosoftImagine;
2. Среды разработки семейства MicrosoftVisualStudio, – лицензия по подписке MicrosoftImagine;

Программа составлена в соответствии с требованиями ОС ННГУ 01.03.02 Прикладная математика и информатика.

Авторы: К.т.н., доцент И.Б. Мееров

К.т.н., доцент А.В. Сысоев

Заведующий кафедрой МОСТ Р.Г. Стронгин

Программа одобрена на заседании методической комиссии института информационных технологий, математики и механики

от 30.11.2022 года, протокол № 3