

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  
от 14.12.2021 г.  
протокол № 4

## **Рабочая программа дисциплины**

**Основы программирования  
Basics of Programming**

---

Уровень высшего образования  
**бакалавриат**

---

Направление подготовки / специальность  
**02.03.02 Фундаментальная информатика и информационные технологии**  
*(указывается код и наименование направления подготовки / специальности)*

---

Направленность образовательной программы  
**Общий профиль**

---

Форма обучения  
**очная**

---

Нижний Новгород  
2022

## 21. Место дисциплины в структуре ОПОП

Данная дисциплина читается на первом курсе бакалавриата (Б1.О.13 – обязательная часть) направления подготовки 02.03.02 «Фундаментальная информатика и информационные технологии». Для изучения дисциплины не требуется каких-либо специальных знаний, за исключением знаний, не выходящих за рамки школьной программы.

Дисциплина относится к обязательной части ООП по направлению подготовки 02.03.02 «Фундаментальная информатика и информационные технологии». Дисциплина читается студентам 1 курса в 1 и 2 семестрах, 9 зачетных единиц, 324 часа, зачет по окончании 1 семестра, экзамен по окончании 2 семестра.

The subject is taught in the first course of a bachelor degree (B1.O.13 – mandatory part) directions of preparation 02.03.02 "Fundamental Informatics and information technologies". The discipline does not require any special knowledge, except for knowledge that does not go beyond the school curriculum.

Дисциплина является основой для последующего изучения других курсов по математике и информатике, предусмотренных программой направления подготовки 02.03.02 «Фундаментальная информатика и информационные технологии».

№ 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.13 «Основы программирования» относится к обязательной части ООП направления подготовки 02.03.02 «Фундаментальная информатика и информационные технологии».

### Цель освоения дисциплины

Дисциплина «Основы программирования» является первой частью двухгодичного курса по различным аспектам программирования, **общей целью которого** является подготовка высококвалифицированных разработчиков сложных программных систем моделирования объектов и явлений реального мира, управления экономико-социальными и производственными процессами, а также решения других задач автоматизации, научных исследований и проектирования на основе применения современной вычислительной техники.

Данная дисциплина **преследует цель** систематического изучения следующих аспектов:

- общие вопросы создания программ, включая основные этапы процесса разработки и используемые средства;
- краткие сведения о среде исполнения программ;
- краткие сведения об инструментах программирования;
- основные элементы и принципы построения языков программирования высокого уровня на примере языков С и С++;
- базовые алгоритмы и основы алгоритмизации с примерами на языке С;
- различные способы описания моделей объектов предметной области с помощью конструирования типов данных с использованием средств С и С++;
- вопросы динамического управления памятью и работы с файлами с использованием средств С и С++;

- основы технологии объектно-ориентированного программирования на примере C++; 3
- углубленные элементы технологии объектно-ориентированного программирования: наследование, виртуальные методы, шаблоны функций и классов.

## 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	
<b>ОПК-3.</b> Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, информационного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	ОПК-3.1. Знает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей	<b>ЗНАТЬ</b> основные идеи, лежащие в основе современных языков программирования и основные составляющие современного языка программирования на примере C и C++. / the main ideas underlying the modern programming languages and the main components of the modern programming language on the example of C and C++. принципы реализации технологий разработки программного обеспечения: структурное, модульное и объектно-ориентированное программирование – в языках программирования C и C++; / the principles of realization of technology of software development: structural, modular and object-oriented programming in C++, IP методы конструирования новых типов данных (C, C++) / methods of constructing new data types (C, C++). методы и приемы работы с динамической памятью (C, C++). / methods and techniques for working with dynamic memory (C, C++). методы работы с внешней памятью (C, C++) / methods of working with external memory (C, C++).	Собеседование (экзамен)
	ОПК-3.2. Умеет соотносить знания в области программирования, интерпретацию прочитанного, определять и создавать информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем.	<b>УМЕТЬ</b> разрабатывать и отлаживать программы средней сложности с использованием языков программирования C, C++. / develop and debug programs of medium complexity using C, C++ programming languages. использовать современные инструментальные и вычислительные средства. / the use of modern instrumental and computational resources.	Лабораторная работа
	ОПК-3.3. Имеет практический опыт применения разработки	<b>ВЛАДЕТЬ</b> современными интегрированными средами разработки программ. / modern integrated development environments programs	Лабораторная работа

	программного обеспечения.	навыками реализации, тестирования и отладки программных систем средней сложности / skills of implementation, testing and debugging of software systems of medium complexity	
--	---------------------------	---	--

### 3. Структура и содержание дисциплины

#### 3.1. Трудоемкость дисциплины

	Очная форма обучения
<b>Общая трудоемкость</b>	<b>9 ЗЕТ</b>
<b>Часов по учебному плану</b>	<b>324</b>
<b>в том числе</b>	
<b>аудиторные занятия (контактная работа):</b>	<b>195</b>
- занятия лекционного типа	64
- занятия семинарского типа	64
- занятия лабораторного типа	64
- текущий контроль (КСР)	3
<b>самостоятельная работа</b>	<b>93</b>
<b>Промежуточная аттестация – экзамен/зачет</b>	<b>36</b>

#### 3.2. Содержание дисциплины

Наименование и краткое содержание 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	Занятия лабораторного типа	Всего	
Первый семестр						
Введение в предмет. Структура и содержание курса. Литература. Introduction to the subject. Structure and content of the course. Literature.	8	2	2	2	6	3
Решение задач с использованием вычислительной техники. Solving problems using computers.	11	2	2	2	6	3
Современная система разработки программного обеспечения. Modern software development system	11	2	2	2	6	3
Среда исполнения программ. Программа в среде MicrosoftWindows. The program execution environment. The program in the Microsoft Windows environment	11	2	2	2	6	3
Основные понятия языков программирования. Синтаксис, семантика, формальные способы описания	11	2	2	2	6	3

языков программирования. Basic concepts of programming languages. Syntax, semantics, formal ways of describing programming languages.						
Типы данных, способы и механизмы управления данными. Data types, ways and mechanisms of data management.	11	2	2	2	6	4
Программа на языке C. Методы и основные этапы трансляции. Program in C language. Methods and basic stages of translation.	11	2	2	2	6	4
Структурное программирование и операторы языка C. Structured programming and C language operators	14	3	3	3	9	4
Конструирование новых типов данных. Designing new types of data.	14	3	3	3	9	4
Модульное программирование. Modular programming	14	3	3	3	9	4
Элементы анализа и разработки алгоритмов. Elements of analysis and development of algorithms	14	3	3	3	9	4
Методы работы с внешней памятью. Файлы. Methods of working with external memory. Files.	14	3	3	3	9	4
Динамическое управление памятью. Dynamic memory management.	15	3	3	3	9	4
Текущий контроль (КСР)	1				1	
Промежуточная аттестация – зачет						
Второй семестр						
Введение в объектно-ориентированное программирование / Introduction to object-oriented programming	8	2	2	2	6	2
Классы и объекты / Classes and objects	12	2	2	2	6	4
Конструкторы и деструктор / Constructors and destructor	12	2	2	2	6	4
Обработка исключений / Exception handling	12	2	2	2	6	4
Перегрузка операций / Overloading operations	16	3	3	3	9	4
Пример. Класс CDate / Example. Class CDate	16	3	3	3	9	4
Наследование и иерархия классов / Inheritance and Class Hierarchy	16	3	3	3	9	4
Специальные поля и методы классов / Special fields and methods of classes	16	3	3	3	9	4
Виртуальные методы. Абстрактные виртуальные методы и классы / Virtual methods. Abstract virtual methods and classes	16	3	3	3	9	4
Шаблоны. Шаблоны функций и шаблоны классов / Templates. Function templates and class templates	16	3	3	3	9	4
Пример. Класс Container / Example. Container class	17	3	3	3	9	4
Пример. Класс Dictionary / Example. Dictionary class	17	3	3	3	9	4
Текущий контроль (КСР)	2				2	
Промежуточная аттестация – экзамен	36					
Итого	324	64	64	64	195	93

Текущий контроль успеваемости реализуется в форме опросов на занятиях семинарского типа, проведения лабораторных работ. Промежуточная аттестация проходит в традиционных формах (зачет, экзамен).

#### 4. Учебно-методическое обеспечение самостоятельной работы обучающихся

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 лабораторных работ на следующие темы:
  - Лабораторная работа 1. Алгоритмы сортировки и поиска данных. / Laboratory work 1. Algorithms for sorting and searching for data.
  - Лабораторная работа 2. Суммирование рядов. / Laboratory work 2. Summation of series.
  - Лабораторная работа 3. Решение систем алгебраических линейных уравнений. / Laboratory work 3. Solving systems of algebraic linear equations.

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	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые	Продemonстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания,	Продemonстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания, в	Продemonстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме, но	Продemonстрированы все основные умения, решены все основные задачи с отдельными несущественными недочетами,	Продemonстрированы все основные умения, решены все основные задачи. Выполнены все задания, в полном объеме без

	я от ответа	ошибки.	но не в полном объеме.	полном объеме, но некоторые с недочетами.	некоторые с недочетами.	выполнены все задания в полном объеме.	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

вопросы	Код формируемой компетенции
1. Введение в языки программирования. Языки высокого/низкого уровня. Понятия программы, переменной, способа интерпретации, типа данных. Тривиальная программа “Hello, world” на C. / Introduction to programming languages. High / low languages. The concepts of the program, the	ОПК-3

<p>variable, the way of interpretation, the type of data. Trivial program "Hello, world" on C.</p>	
<p>2. Структура программы на C, синтаксис. Операторы ввода-вывода. Типы данных. Операторы ветвления. Программы: / Structure and syntax of the program in C. Operators of input-output. Data types. Branching operators. Programs:</p> <ul style="list-style-type: none"> <li>Определить сумму налога за участок треугольной формы (задаётся ставка налога). / Determine the amount of tax for a triangular plot (the tax rate is set).</li> <li>Определить вид треугольника по заданным длинам сторон. Обеспечение контроля ввода. / Determine the shape of the triangle for the given lengths of the sides. Provide input control.</li> </ul>	ОПК-3
<p>3. Обсуждение правильного использования операторов if...then и if...then...else. Оператор множественного выбора. Циклы. Контроль ввода (циклы с постусловием). Программы: / Discussing the correct use of if ... then and if ... then ... else statements. Multiple choice operator. Cycles. Input control (loops with postcondition). Programs:</p> <ul style="list-style-type: none"> <li>Реализовать калькулятор с поддержкой операций: +, -, *, /. / Implement a calculator with support for operations: +, -, *, /.</li> <li>Программа «угадай число» / The "guess number" program</li> </ul>	ОПК-3
<p>4. Массивы (синтаксис, примеры объявления, индексация). Генерация псевдослучайных данных. Прерывание циклов (break, continue). Бинарный поиск, простейшая сортировка. Программы: / Arrays (syntax, ad examples, indexing). Pseudo-random data generation. Interrupting the cycles (break, continue). Binary search, the simplest sorting. Programs:</p> <ul style="list-style-type: none"> <li>Вывести на экран коды всех кодов символов A..Z (необходимо использовать массив для хранения кодов, в качестве индекса использовать соответствующий символ) / Display the codes of all A..Z character codes (use the array to store the codes, use the corresponding symbol as the index)</li> <li>Найти min и max элементы в массиве и их позицию. Найти заданный элемент в массиве и его позицию. / Find the min and max elements in the array and their position. Find the given element in the array and its position.</li> <li>Sorting. Find the element in the ordered array.</li> <li>Удаление/добавление элемента в массив (упорядоченный, неупорядоченный) / Deleting / adding an element to an array (ordered, unordered)</li> </ul>	ОПК-3
<p>5. Изучение следующих сортировок (оценка эффективности): / Study of the following sorts (effectiveness evaluation):</p> <ul style="list-style-type: none"> <li>Сортировка пузырьком. / Bubble sort</li> <li>Сортировка выбором. / Selection sort</li> <li>Сортировка вставками. / Insertion sort</li> <li>Сортировка слиянием / Merge sort</li> </ul>	ОПК-3
<p>6. Подпрограммы (назначение, виды). Передача параметров по значению и по ссылке. Рекурсия. Программы: / Subroutines (purpose, types). Passing parameters by value and by reference. Recursion. Programs:</p> <ul style="list-style-type: none"> <li>Реализовать функцию вычисления расстояния между двумя точками на плоскости / Implement a function to calculate the distance between two points on a plane.</li> <li>Дан массив, реализовать функцию, которая выводит индексы элементов массива, соответствующие возрастающему порядку элементов / Given an array, implement a function that prints the indices of the elements of the array, corresponding to the increasing order of the elements.</li> </ul>	ОПК-3
<p>7. Передача параметров в подпрограммы (параметры-константы, параметры без типа, массивы и строки открытого типа). Программы: / Passing parameters to subroutines (parameters-constants, parameters without type, arrays and strings of an open type). Programs:</p> <ul style="list-style-type: none"> <li>Функция поиска элемента в массиве / The function of finding an element in an array.</li> <li>Функция сортировки массива произвольного размера / The function of sorting an array of arbitrary size.</li> </ul>	ОПК-3



8. Процедурный тип данных. Пример использования процедурного типа на примере сортировки (функция сравнения элементов) / Procedural data type. An example of using a procedural type in the sorting example (element comparison function)	ОПК-3
9. Конструирование составных типов данных. Модули. Пример реализации модуля для работы с комплексными числами / Construction of composite data types. Modules. An example of implementing a module for working with complex numbers.	ОПК-3
10. Файловый ввод-вывод (основные понятия, организация файлового ввода-вывода в С). Решение задач с использованием файлового ввода-вывода: / File input / output (basic concepts, organization of file input/output in C). Solving problems using file input/output: <ul style="list-style-type: none"> <li>The file contains two-dimensional coordinates of the points defining the vertices of the angles of the polygon. Implement a subroutine that determines whether a polygon is convex or not.</li> </ul>	ОПК-3

9

### 5.2.2. Темы лабораторных работ

- Лабораторная работа 1. Алгоритмы сортировки и поиска данных. / Laboratory work 1. Algorithms for sorting and searching for data.
- Лабораторная работа 2. Суммирование рядов. / Laboratory work 2. Summation of series.
- Лабораторная работа 3. Решение систем алгебраических линейных уравнений. / Laboratory work 3. Solving systems of algebraic linear equations.

### 5.2.3. Примеры экзаменационных билетов

Национальный исследовательский Нижегородский государственный университет  
им. Н.И. Лобачевского  
Институт ИТММ  
Кафедра математического обеспечения и суперкомпьютерных технологий  
Дисциплина «Основы программирования»

#### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

- Программа на языке С. Объявления, операторы, комментарии, директивы компилятору,
- Структура программы на языке С. Примеры.

- Program in the language C. Declarations, operators, comments, directives to the compiler,
- The structure of the program in C. Examples.

Зав. кафедрой \_\_\_\_\_ Р.Г. Стронгин  
Экзаменатор \_\_\_\_\_ К.А. Баркалов

Национальный исследовательский Нижегородский государственный университет  
им. Н.И. Лобачевского  
Институт ИТММ  
Кафедра математического обеспечения и суперкомпьютерных технологий  
Дисциплина «Основы программирования»

## ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 2

1. Понятие типа данных. Логический тип и логические выражения в языке C.
2. Перечислимый тип в языке C. Операции над объектами этих типов. Примеры использования.

1. The notion of a data type. Logical type and logical expressions in C.
2. An enumerated type in the language C. Operations on objects of these types. Examples of using.

Зав. кафедрой \_\_\_\_\_ Р.Г. Стронгин

Экзаменатор \_\_\_\_\_ К.А. Баркалов

**6. Учебно-методическое и информационное обеспечение дисциплины**

а) основная литература:

- B. Kernighan, D. Ritchie, The C Programming Language. – <https://link.springer.com>
- B. Stroustrup: The C++ Programming Language. – <https://link.springer.com>

•

б) дополнительная литература:

- H. Schildt C: The Complete Reference. – <https://link.springer.com>
- H. Schildt C++: The Complete Reference. – <https://link.springer.com>

**7. Материально-техническое обеспечение дисциплины**

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой (лекционного и семинарского типа), 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.

Программа составлена в соответствии с требованиями ФГОС ВО /ОС ННГУ \_\_\_\_\_.

Автор \_\_\_\_\_ К.А. Баркалов

Рецензент \_\_\_\_\_

Зав. кафедрой \_\_\_\_\_ Р.Г. Стронгин

Программа одобрена на заседании методической комиссии института информационных технологий, математики и механики от 01.12.2021 года, протокол № 2.

