

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 Ученого совета ННГУ

протокол № 10 от 02.12.2024 г.

**Рабочая программа дисциплины**

Программирование на C++. Часть 1

---

Уровень высшего образования

Магистратура

---

Направление подготовки / специальность

02.04.02 - Фундаментальная информатика и информационные технологии

---

Направленность образовательной программы

Искусственный интеллект

---

Форма обучения

очная

---

г. Нижний Новгород

2025 год начала подготовки

## 1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.03.02 Программирование на C++. Часть 1 относится к части, формируемой участниками образовательных отношений 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
ПК-8: Способен к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности	ПК-8.1: Знает методику разработки новых алгоритмических, методических и технологических решений ПК-8.2: Умеет применять полученные знания для разработки новых алгоритмических, методических и технологических решений ПК-8.3: Имеет практический опыт составления технического задания на разработку информационной системы	ПК-8.1: Знает основы ИТ в области КС и КГ и имеет навыки анализа современного состояния науки и информационных технологий в этой области.  ПК-8.2: Имеет навыки проектирования и разработки и развития ИТ- решений на основе анализа современного состояния науки и информационных технологий в области КС и КГ  ПК-8.3: Владеет навыками управления разработкой и развитием ИТ- решений на основе анализа современного состояния науки и информационных технологий в области КС и КГ	Практическая задача	Зачёт: Контрольные вопросы

## 3. Структура и содержание дисциплины

### 3.1 Трудоемкость дисциплины

	очная
Общая трудоемкость, з.е.	3
Часов по учебному плану	108
в том числе	

<b>аудиторные занятия (контактная работа):</b>	
<b>- занятия лекционного типа</b>	<b>32</b>
<b>- занятия семинарского типа (практические занятия / лабораторные работы)</b>	<b>32</b>
<b>- КСР</b>	<b>1</b>
<b>самостоятельная работа</b>	<b>43</b>
<b>Промежуточная аттестация</b>	<b>0</b> <b>Зачёт</b>

### 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) Введение. Функции, переменные, типы. Операторы и выражения. Компиляция, заголовочные файлы.	7	2	2	4	3
2) Указатели и ссылки. Параметры и возвращаемые значения функций. Арифметика указателей.	10	3	3	6	4
3) Структуры и классы. ООП, конструкторы, деструкторы, операторы. Инкапсуляция.	10	3	3	6	4
4) Шаблоны и STL. Контейнеры, алгоритмы, итераторы. Функторы и лямбды.	10	3	3	6	4
5) Динамическое управление памятью. Умные указатели. RAII.	10	3	3	6	4
6) Наследование. Виртуальные функции, абстрактные классы. Исключения.	10	3	3	6	4
7) Паттерны проектирования.	10	3	3	6	4
8) Расположение объектов в памяти, выравнивание.	10	3	3	6	4
9) Отладка, GDB, core dumps.	10	3	3	6	4
10) Строки. Кодировки и регулярные выражения.	10	3	3	6	4
11) Move-семантика. rvalue-ссылки, variadic templates, perfect forwarding.	10	3	3	6	4
Аттестация	0				
КСР	1				1
Итого	108	32	32	65	43

#### Содержание разделов и тем дисциплины

- 1) Введение. Функции, переменные, типы. Операторы и выражения. Компиляция, заголовочные файлы.
- 2) Указатели и ссылки. Параметры и возвращаемые значения функций. Арифметика указателей.

- 3) Структуры и классы. ООП, конструкторы, деструкторы, операторы. Инкапсуляция.
- 4) Шаблоны и STL. Контейнеры, алгоритмы, итераторы. Функторы и лямбды.
- 5) Динамическое управление памятью. Умные указатели. RAII.
- 6) Наследование. Виртуальные функции, абстрактные классы. Исключения.
- 7) Паттерны проектирования.
- 8) Расположение объектов в памяти, выравнивание.
- 9) Отладка, GDB, core dumps.
- 10) Строки. Кодировки и регулярные выражения.
- 11) Move-семантика. rvalue-ссылки, variadic templates, perfect forwarding

#### **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.

Дополнительная литература:

1. Павловская Т. «Программирование на языке C++».

<http://www.intuit.ru/studies/courses/626/482/info>

Программное обеспечение и Интернет-ресурсы:

1. <https://msdn.microsoft.com/ru-ru/dn308572.aspx>
2. <https://gitlab.com/moskalenkoviktor>

#### **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 - Практическая задача) для оценки сформированности компетенции ПК-8:**

Практическая часть курса предлагает к решению 70 задач различного направления и сложности (сложность пропорциональна числу баллов), с решением их значительной части (не менее 40%), закрепляя положения всей теоретической части.

##### **Структура решаемых задач**

В каждой задаче дано как минимум 3 файла:

- CMakeLists.txt - cmake-файл для сборки задачи.
- test.cpp - тесты для автоматической проверки задачи.
- \*.h - .h файл, в котором нужно реализовать то, что требуется в условии задачи.

Разрешено изменять только файлы, содержащие решение задачи, в противном случае система не примет Вашу посылку на тестирование. Когда Вы делаете посылку, система запускает test.cpp, и в случае успешного завершения засчитывается количество баллов, соответствующее сложности задачи.

Некоторые задачи test.cpp вообще не содержат тестов, или содержат не все тесты, которые должны быть на сервере. Об этом всегда сообщается в условии задачи. В этом случае необходимо самостоятельно написать тесты, тестирующие всю реализованную функциональность. См. также ресурс:

<https://gitlab.com/moskalenkoviktor>

### Примеры задач:

1. Задача для знакомства с тестирующей системой. Реализовать функцию Multiply, которая перемножает 2 числа, в файле multiplication.h
2. Реализуйте функцию быстрого возведения в степень BinPow(a, b, c), которая вычисляет  $a^b \pmod c$ . Количество операций должно быть пропорционально двоичному логарифму b.
3. Реализовать функцию Range(from, to, step), аналог функции range из python2- возвращает вектор всех чисел из диапазона [from, to) с шагом step. Примеры в тест-коде.
4. Реализовать функцию Unique, принимающую отсортированный по неубыванию вектор чисел и возвращает вектор, в котором оставлены только различные элементы.
5. Реализовать функцию Split(string, delimiter), которая разбивает строку string на подстроки по заданному разделителю delimiter. Например, Split("just a test", " ") = {"just", "a", "test"}. Обрабатывать и разделители идущие подряд.
6. Реализуйте функцию EvaluateExpression, которая вычисляет целочисленное выражение в обратной польской записи (операции +, \*, -, целые числа разделенные пробелами).
7. Реализуйте функцию CreateYandexForecaster, которая возвращает текущую погоду, используя API Яндекс.Погоды. Данная функция принимает ключ для доступа к API и url, к которому нужно делать запросы (используется в тестах). Интерфейсы приведены в файле weather.h, реализуйте все необходимое в weather.cpp. Реализация интерфейса IForecaster позволяет принимать ширину-долготу, которые нужно использовать при запросе.
8. Написать telegram-бота, используя библиотеку РОСО. В задании будет оцениваться дизайн и реализация клиента для общения с серверами telegram. Реализация 3 дополнительных функций – до 400 премиальных баллов
9. По материалам семинара написать рейтрейсер, с помощью которого можно рендерить простые полигональные сцены. Основная часть задачи будет описана на семинаре. Презентация семинара лежит в файлах задачи. В составе задачи есть 3 вспомогательные задачи:  
<https://gitlab.com/moskalenkoviktor/shad-cpp0-nn/-/tree/master/raytracer-geom>  
<https://gitlab.com/moskalenkoviktor/shad-cpp0-nn/-/tree/master/raytracer-reader>  
<https://gitlab.com/moskalenkoviktor/shad-cpp0-nn/-/tree/master/raytracer-debug>

Полный список задач см. на ресурсе: <https://gitlab.com/moskalenkoviktor>

### Критерии оценивания (оценочное средство - Практическая задача)

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. Всего набрано не менее 40% от суммы баллов по всем задачам.
не зачтено	Выполнены не все лабораторные работы или выполнены не в полном объеме (представлено не полное описание этапов выполнения заданий, программа работает некорректно, результаты работы не представлены преподавателю). Всего набрано менее 40% от суммы баллов по всем задачам.

## 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	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 отдельным и 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 базовых	При решении стандартных	Имеется минимальн	Продemonстрированы	Продemonстрированы	Продemonстрированы	Продemonстрированы

	навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	задач не продемонстрир ованы базовые навыки. Имели место грубые ошибки	ый набор навыков для решения стандартны х задач с некоторым и недочетами	базовые навыки при решении стандартны х задач с некоторым и недочетами	базовые навыки при решении стандартны х задач без ошибок и недочетов	навыки при решении нестандарт ных задач без ошибок и недочетов	творческий подход к решению нестандартны х 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».
	удовлетворитель но	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	неудовлетворите льно	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 - Контрольные вопросы) для оценки сформированности компетенции ПК-8

1. Какие существуют контейнеры STL? Каковы сложность операций работы с ними? Примеры.
2. Типы итераторов. Связь контейнеров STL и итераторов.
3. Примеры использования decltype и auto, основные правила вывода типов.
4. Почему невозможно копировать объект типа std::unique\_ptr? Примеры.
5. Что такое std::weak\_ptr? Зачем он нужен? Примеры.
6. Типичные шаблоны проектирования с использованием интеллектуальных указателей.
7. Оптимизация классов с использованием семантики перемещения.
8. Реализация семантики перемещения в контейнерах STL.
9. Методы по умолчанию, генерируемые в классах.
10. Списки инициализации. Плюсы и минусы.

11. Функциональные объекты, функции, лямбда-выражения, функторы. Плюсы и минусы.
12. Является ли функция объявленная с модификатором const потокобезопасной? Примеры.
13. Параллельные программы на C++. Подход к разработке.
14. Реализация std::tuple с использованием различных стандартов языка C++.

### Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
зачтено	Студент ответил на большую часть вопросов возможно с незначительными недочетами. Студент работал на практических занятиях и набрал не менее 40% от суммы баллов по всем задачам.
не зачтено	При ответе студент допускает грубые ошибки в основном материале. Студент не работал на практических занятиях и набрал менее 40% от суммы баллов по всем задачам.

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

Основная литература:

1. Язык программирования C++ для профессионалов / Страуструп Б. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=663690&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. Операционные системы семейства Microsoft Windows, лицензия по подписке Microsoft Imagine.
2. Среда 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 02.04.02 - Фундаментальная информатика и информационные технологии.



Автор(ы): Москаленко Виктор Алексеевич.

Заведующий кафедрой: Золотых Николай Юрьевич, доктор физико-математических наук.

Программа одобрена на заседании методической комиссии от 02.12.2024, протокол № 5.