

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский Нижегородский государственный университет им. Н.И.  
Лобачевского»  
Институт информационных технологий,  
математики и механики  
Кафедра математического обеспечения и суперкомпьютерных технологий

УТВЕРЖДЕНО  
решением Ученого совета ННГУ  
протокол от 30 ноября 2022 г. № 13

Рабочая программа дисциплины  
«Язык C++»

---

Уровень высшего образования  
магистратура

---

Направление подготовки  
01.04.02 «Прикладная математика и информатика»

---

Профиль подготовки  
«Анализ данных в прикладных областях»

---

Квалификация выпускника  
Магистр

---

Форма обучения  
Очная

---

Нижний Новгород  
2023 год

## 1. Место дисциплины в структуре ООП

№ варианта	Место дисциплины в учебном плане образовательной программы	Стандартный текст для автоматического заполнения в конструкторе РПД
1	Блок 1. Дисциплины (модули) по выбору	Дисциплина Б1.В.ДВ.01.01 «Язык C++» относится к дисциплинам по выбору части, формируемой участниками отношений, ООП направления подготовки "01.04.02 Прикладная математика и информатика".

## 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**	
ПК-5. Способен разрабатывать и применять математические методы, системное и прикладное программное обеспечение для решения задач научной деятельности	ПК-5.1. Знает типовые математические методы и методологии разработки системного и прикладного программного обеспечения для решения задач научной деятельности.	<b>Знает</b> основы ИТ в области КС и КГ и имеет навыки разработки системного и прикладного программного обеспечения для решения задач научной деятельности.	собеседование,
	ПК-5.2. Умеет применять типовые математические методы и методологии разработки системного и прикладного программного обеспечения для решения задач научной деятельности	Умеет применять навыки проектирования и разработки и развития ИТ-решений на основе разработки системного и прикладного программного обеспечения для решения задач научной деятельности	задания
	ПК-5.3. Имеет навыки разработки системного и прикладного программного обеспечения для решения задач научной деятельности	<b>Имеет</b> навыки проектирования и разработки и развития ИТ-решений на основе разработки системного и прикладного программного обеспечения для решения задач научной деятельности	задания
ПК-12. Способен разрабатывать и применять математические методы, системное и прикладное программное обеспечение для решения задач производственно-технологической деятельности	ПК-12.1. Знает основные методы разработки математических методов, системного и прикладного программного обеспечения для решения задач производственно-технологической деятельности.	<b>Знает</b> основы ИТ в области КС и КГ и имеет навыки разработки системного и прикладного программного обеспечения для решения задач производственно-технологической деятельности	собеседование,
	ПК-12.2. Умеет оценивать трудоемкость разработки программных средств для решения задач производственно-технологической деятельности	Умеет применять навыки проектирования и разработки и развития ИТ-решений на основе разработки системного и прикладного программного обеспечения для решения задач производственно-технологической деятельности	задания

	ПК-12.3. Имеет навыки разработки системного программного обеспечения для решения задач производственно-технологической деятельности	<b>Имеет</b> навыки проектирования и разработки и развития ИТ-решений на основе разработки системного и прикладного программного обеспечения для решения задач производственно-технологической деятельности	задания
--	---	---	---------

### 3. Структура и содержание дисциплины

#### 3.1. Трудоемкость дисциплины

	Очная форма обучения
<b>Общая трудоемкость</b>	<b>7 ЗЕТ</b>
<b>Часов по учебному плану</b>	<b>252</b>
<b>В том числе</b>	
<b>Контактная работа:</b>	<b>33</b>
- Занятия лекционного типа	<b>32</b>
- Занятия семинарского типа	<b>32</b>
- Текущий контроль (КСР)	<b>2</b>
<b>Самостоятельная работа</b>	<b>150</b>
<b>Промежуточная аттестация–зачет</b>	<b>36</b>

#### 3.2. Содержание дисциплины

Наименование и краткое содержание разделов и тем дисциплины	Всего( час.)	В том числе				
		Контактная работа (работа во взаимодействии с преподавателем), часы. Из них				
		Занятия лекционного типа	Занятия семинарского типа	Занятия лабораторного типа	Всего	Саостоятельная работа
1) Введение. Функции, переменные, типы. Операторы и выражения. Компиляция, заголовочные файлы.	14	2	2		4	10
2) Указатели и ссылки. Параметры и возвращаемые значения функций. Арифметика указателей.	14	2	2		4	10
3) Структуры и классы. ООП, конструкторы, деструкторы, операторы. Инкапсуляция.	20	4	4		8	12
4) Шаблоны и STL. Контейнеры, алгоритмы, итераторы. Функторы и лямбды.	25	4	4		8	17
5) Динамическое управление памятью. Умные указатели. RAII.	16	2	2		4	12
6) Наследование. Виртуальные функции, абстрактные классы. Исключения.	16	2	2		4	12
7) Паттерны проектирования.	25	4	4		8	17
8) Расположение объектов в памяти, выравнивание.	19	2	2		4	15
9) Отладка, GDB, core dumps.	23	4	4		8	15

10) Строки. Кодировки и регулярные выражения.	19	2	2		4	15
11) Move-семантика. rvalue-ссылки, variadic templates, perfect forwarding	23	4	4		8	15
Текущий контроль(КСР)	2				2	
Промежуточная аттестация–экзамен	36					
<b>Итого</b>	252	32	32		66	150

Текущий контроль успеваемости реализуется в формах опросов на занятиях семинарского типа. Промежуточная аттестация проходит в традиционной форме – экзамен.

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

Для выполнения программы самостоятельной работы дисциплины «Язык C++» достаточно: (а) самостоятельной проработки лекционного и дополнительного материала и (б) выполнить практическую часть дисциплины.

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 обычно как минимум 3 файла:

- CMakeLists.txt --- stake-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>

Раздел дисциплины	Направления лабораторных работ
Rvalue-ссылки и Семантика перемещений	1) Освоение использования семантики перемещения.
Лямбда - выражения.	1) Освоение использования функциональных объектов и лямбда - выражений.
Параллельные вычисления.	1) Освоение подходов к разработке параллельных программ с использованием потоков C++. 2) Освоение лучших практик использования примитивов синхронизации C++.
Метапрограммирование.	1) Реализация шаблонных контейнеров с количеством элементов, и известными на этапе компиляции. 2) Освоение типовых техник метапрограммирования.

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 (индикатор достижения компетенций)	Шкала оценивания сформированности компетенций						
	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	Незачтено		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 грубых ошибок.	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько грубых 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, превышающий программу подготовки.
<u>Умения</u>	Отсутствия минимальных умений. Невозможность оценить умения вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки.	Продemonстрированы основные умения. Решены типовые задачи с грубыми ошибками. Выполнены все задания, но не в полном объеме.	Продemonстрированы все основные умения. Решены все основные задачи с грубыми 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>	Отсутствия 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.	Продemonстрировано творческий подход к решению нестандартных задач.

**Шкала оценки при промежуточной аттестации**

Оценка		Уровень подготовки
	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»

зачтено	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»
незачтено	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 Примеры контрольных вопросов для проверки ПК-5, ПК-12

Вопросы	Код компетенции
1) Какие существуют контейнеры STL? Каковы сложность операций работы с ними? Примеры.	ПК-5
2) Типы итераторов. Связь контейнеров STL и итераторов.	ПК-5
3) Примеры использования <code>decltype</code> и <code>auto</code> , основные правила вывода типов.	ПК-5
4) Почему невозможно копировать объект типа <code>std::unique_ptr</code> ? Примеры.	ПК-5
5) Что такое <code>std::weak_ptr</code> ? Зачем он нужен? Примеры.	ПК-5
6) Типичные шаблоны проектирования с использованием интеллектуальных указателей.	ПК-5
7) Оптимизация классов с использованием семантики перемещения.	ПК-5
8) Реализация семантики перемещения в контейнерах STL.	ПК-12
9) Методы по умолчанию, генерируемые в классах.	ПК-12
10) Списки инициализации. Плюсы и минусы.	ПК-12
11) Функциональные объекты, функции, лямбда - выражения, функторы. Плюсы и минусы.	ПК-12
12) Является ли функция объявленная с модификатором <code>const</code> потокобезопасной? Примеры.	ПК-12
13) Параллельные программы на C++. Подход к разработке.	ПК-12
14) Реализация <code>std::tuple</code> с использованием различных стандартов языка C++.	ПК-12

### 5.2.2. Типовые задания для оценки сформированности компетенции ПК-5, ПК-12

Решение задач практики из предложенного списка до набора установленного числа баллов (40% от суммы баллов по всем задачам).

Примеры содержания задач
1. Задача для знакомства с тестирующей системой. Реализовать функцию <code>Multiply</code> , которая перемножает 2 числа, в файле <code>multiplication.h</code>
2. Реализуйте функцию быстрого возведения в степень <code>BinPow(a, b, c)</code> , которая вычисляет $a^b \pmod c$ . Количество операций должно быть пропорционально двоичному логарифму $b$ .
3. Реализовать функцию <code>Range(from, to, step)</code> , аналог функции <code>range</code> из <code>python2</code> - возвращает вектор всех чисел из диапазона $[from, to)$ с шагом <code>step</code> . Примеры в тест-коде.
4. Реализовать функцию <code>Unique</code> , 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"}`. Обработать и разделители идущие подряд.
  6. Реализуйте функцию `EvaluateExpression`, которая вычисляет целочисленное выражение в обратной польской записи (операции `+`, `*`, `-`, целые числа разделенные пробелами).
  60. 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.
  61. Написать telegram-бота, используя библиотеку `POCO`. В задании будет оцениваться дизайн и реализация клиента для общения с серверами telegram. Реализация 3 дополнительных функций – до 400 премиальных баллов
  69. 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 ресупсе: <https://gitlab.com/moskalenkoviktor>

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

### **а) основная литература:**

1. Страуструп Б. «Язык программирования C++ для профессионалов». <http://www.intuit.ru/studies/courses/98/98/info>
2. Павловская Т. «Программирование на языке C++». <http://www.intuit.ru/studies/courses/626/482/info>

### **б) дополнительная литература:**

1. Фридман А. «Язык программирования C++». <http://www.intuit.ru/studies/courses/17/17/info>

### **в) программное обеспечение и Интернет-ресурсы:**

<https://msdn.microsoft.com/ru-ru/dn308572.aspx>

## **7. Материально-техническое обеспечение дисциплины**

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой (лекционного типа), оснащенные оборудованием и техническими средствами обучения: компьютерной техникой с возможностью подключения к сети «Интернет».

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 ОС ННГУ по направлению 01.04.02 Прикладная математика и информатика.

Автор \_\_\_\_\_ Золотых Н.Ю.

Рецензент(ы) \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_ Золотых Н.Ю.

Программа одобрена на заседании методической комиссии института информационных технологий, математики и механики от 30 ноября 2022 года, протокол № 3.