

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

Язык C++

Уровень высшего образования
Магистратура

Направление подготовки / специальность
01.04.02 - Прикладная математика и информатика

Направленность образовательной программы
Анализ данных в прикладных областях

Форма обучения
очная

г. Нижний Новгород

2025 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.01.01 Язык С++ 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
<i>ПК-12: Способен разрабатывать и применять математические методы, системное и прикладное программное обеспечение для решения задач производственно-технологической деятельности</i>	<p><i>ПК-12.1: Знает основные методы разработки математических методов, системного и прикладного программного обеспечения для решения задач производственно-технологической деятельности</i></p> <p><i>ПК-12.2: Умеет оценивать трудоемкость разработки программных средств для решения задач производственно-технологической деятельности</i></p> <p><i>ПК-12.3: Имеет навыки разработки системного программного обеспечения для решения задач производственно-технологической деятельности</i></p>	<p><i>ПК-12.1: Знает основы ИТ в области КС и КГ и имеет навыки разработки системного и прикладного программного обеспечения для решения задач производственно-технологической деятельности</i></p> <p><i>ПК-12.2: Умеет применять навыки проектирования и разработки и развития ИТ-решений на основе разработки системного и прикладного программного обеспечения для решения задач производственно-технологической деятельности</i></p> <p><i>ПК-12.3: Имеет навыки проектирования и разработки и развития ИТ-решений на основе разработки системного и прикладного программного обеспечения для решения задач производственно-технологической деятельности</i></p>	<i>Задачи</i>	<i>Зачёт: Задачи</i>
<i>ПК-5: Способен</i>	<i>ПК-5.1: Знает типовые</i>	<i>ПК-5.1:</i>	<i>Задачи</i>	

разрабатывать и применять математические методы, системное и прикладное программное обеспечение для решения задач научной деятельности	математические методы и методологии разработки системного и прикладного программного обеспечения для решения задач научной деятельности ПК-5.2: Умеет применять типовые математические методы и методологии разработки системного и прикладного программного обеспечения для решения задач научной деятельности ПК-5.3: Имеет навыки разработки системного и прикладного программного обеспечения для решения задач научной деятельности	Знает основы ИТ в области КС и КГ и имеет навыки разработки системного и прикладного программного обеспечения для решения задач научной деятельности. ПК-5.2: Умеет применять навыки проектирования и разработки и развития ИТ-решений на основе разработки системного и прикладного программного обеспечения для решения задач научной деятельности ПК-5.3: Имеет навыки проектирования и разработки и развития ИТ-решений на основе разработки системного и прикладного программного обеспечения для решения задач научной деятельности		Зачёт: Задачи
--	--	---	--	------------------

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очная
Общая трудоемкость, з.е.	6
Часов по учебному плану	216
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	32
- занятия семинарского типа (практические занятия / лабораторные работы)	32
- КСР	2
самостоятельная работа	150
Промежуточная аттестация	0 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	
Введение и основы языка.	14	2	2	4	10
Указатели, ссылки и структуры.	18	4	2	6	12
Классы.	18	4	2	6	12
Шаблоны, STL, lambda-функции.	18	2	4	6	12
Динамическое управление памятью.	18	2	4	6	12
Управление ресурсами.	18	4	2	6	12
Наследование и полиморфизм.	18	2	4	6	12
Обработка ошибок и исключения.	14	2	2	4	10
Компиляция проектов.	14	2	2	4	10
Расположение объектов в памяти. Выравнивание.	16	2	2	4	12
Паттерны проектирования.	16	2	2	4	12
Отладка в GDB.	16	2	2	4	12
Move-семантика.	16	2	2	4	12
Аттестация	0				
КСР	2			2	
Итого	216	32	32	66	150

Содержание разделов и тем дисциплины

1. Введение и основы языка. Стандарты C++. Компиляторы. Переменные, примитивные типы. Арифметические, логические, битовые операторы, операторы сравнения. Ввод и вывод. Функции, циклы, условия. Контейнеры и итераторы.
2. Указатели, ссылки и структуры. Массивы, указатели и ссылки. Операции с указателями. Время жизни объектов. Константность параметров. Структуры, перечисления. Передача параметров.
3. Классы. Структуры, методы, поля, переменная this. Константность методов, их аргументов, полей. Конструкторы, список инициализации, неявные преобразования. Инициализация по умолчанию. Модификаторы доступа.
4. Шаблоны, STL, lambda-функции. Перегрузка функций и шаблонные функции. Шаблонные функции библиотеки STL. Шаблонные классы. Утиная типизация. Итераторы. Строки. lambda-функции, список захвата и мутабельность. Концепты.
5. Динамическое управление памятью. Автоматические переменные. Время жизни объектов. Операторы new и delete. Деструкторы. Копирование и перемещение. Правило пяти. Владение.
6. Управление ресурсами. Rvalue-ссылки и move-семантика. Управление ресурсами. RAII (Resource Acquisition Is Initialization). Умные указатели. Эксклюзивное владение и разделяемое владение. Передача владения.
7. Наследование и полиморфизм. Имена и области видимости. Поиск имени. Наследование. Принцип подстановки Лисков. Модификаторы доступа. Полиморфизм. Динамический полиморфизм. Приведение типов.

8. Обработка ошибок и исключения. Флаги выполнения и коды возврата. Either/Result. Исключения. Иерархия исключений. Гарантии безопасности исключений.
9. Компиляция проектов. Сборка CMake. Модули.
10. Расположение объектов в памяти. Выравнивание. Структуры в памяти, Standard layout type (POD, Plain Old Data). Таблицы виртуальных функций. Выравнивание, операторы alignof и alignas. std::aligned_storage. Упаковка. Указатели на члены.
11. Паттерны проектирования. Порождающие паттерны: фабричный метод (Factory method), абстрактная фабрика (abstract factory), строитель (builder), синглтон (singleton, одиночка). Структурные паттерны: мост (bridge), адаптер (adapter), компоновщик (composite). Поведенческие паттерны (behavioral): команда (command).
12. Отладка в GDB.
13. Move-семантика. 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.

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 50%), закрепляя положения всей теоретической части.

Задачи проверяются автоматически с помощью системы тестирования. В каждой задаче обычно 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.

Помимо проверки корректности кода также проверяется стиль кода.

Среди множества задач есть 3 больших задания, решения которых требует написания программ на несколько сотен строк кода.

Репозиторий с задачами находится по адресу: <https://gitlab.manytask.org/cpp0-nn/public-2024-fall>

При выполнении самостоятельной работы студентам рекомендуется использовать конспекты лекций, а также рекомендуемую в литературу:

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>

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 ПК-12:

Unique

Реализуйте функцию *Unique*, которая принимает отсортированный по не убыванию вектор чисел и возвращает отсортированный вектор, в котором оставлены только различные элементы исходного вектора.

Инвертирование map

Реализуйте функцию *ReverseMap*, которая принимает *std::unordered_map*, содержащий отображение *std::string -> int*, и возвращает обратное представление *int -> std::string*. Гарантируется, что все значения входа различны.

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

Принадлежность точки треугольнику

Реализуйте функцию *IsPointInTriangle(t, p)*, которая проверяет, лежит ли точка *p* внутри (или на границе) треугольника *t*. Гарантируется, что координаты точек не превосходят по модулю 10^9 .

Filter

Реализуйте функцию *FilterEven*, которая принимает вектор целых чисел и оставляет в нем только четные элементы в исходном порядке. Ваш алгоритм должен работать за линейное время.

Критерии оценивания (оценочное средство - Задачи)

Оценка	Критерии оценивания
зачтено	Набрано $\geq 50\%$ баллов за задачи и решено как минимум два больших задания или Набрано $\geq 60\%$ баллов за задачи и решено как минимум одно большое задание
не зачтено	Не выполнены критерии оценки зачтено

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 отдельными 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 некоторыми недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторыми 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 - Задачи) для оценки сформированности компетенции ПК-12

Быстрое возведение в степень

Реализуйте функцию $BinPow(a, b, c)$, которая вычисляет $a^b \bmod c$. Количество операций должно быть пропорционально двоичному логарифму b .

Find Last

Реализуйте шаблон функции $FindLast(first, last, value)$. Функция возвращает итератор на последнее вхождение $value$ в последовательность $[first, last)$ или $last$ в случае, когда такого вхождения нет.

Можете считать, что итераторы по крайней мере двунаправленные.

5.3.2 Типовые задания (оценочное средство - Задачи) для оценки сформированности компетенции ПК-5

Факторизация

Реализуйте функцию $Factorize(n)$, которая раскладывает n на простые делители и возвращает результат в виде вектора пар. Каждая пара имеет вид {делитель, степень вхождения}. Пары должны быть отсортированы по возрастанию делителя. Например,

$Factorize(20) = \{\{2, 2\}, \{5, 1\}\}$

Гарантируется, что $n < 10^{12}$. Количество операций должно быть пропорционально $n^{1/2}$.

Квадратное уравнение

Реализуйте функцию $SolveQuadratic(a, b, c)$, которая находит вещественные корни квадратного уравнения $ax^2 + bx + c = 0$.

В файле `quadratic.h` вы можете найти определение структуры `Roots` (тип возвращаемого результата). В случае, когда уравнение имеет один корень, он должен быть записан в поле `first`. Если же корней два, то функция должна записать их таким образом, что `first < second`.

Критерии оценивания (оценочное средство - Задачи)

Оценка	Критерии оценивания
зачтено	Набрано $\geq 50\%$ баллов за задачи и решено как минимум два больших задания или Набрано $\geq 60\%$ баллов за задачи и решено как минимум одно большое задание
не зачтено	Не выполнены критерии оценки зачтено

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

Основная литература:

1. Воронцова И. О. Программирование на языке высокого уровня C/C++ : учебное пособие / Воронцова И. О., Груздева Л. А., Губанова Т. В. - Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2010. - 111 с. - Книга из коллекции СПбГУТ им. М.А. Бонч-Бруевича - Информатика., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=888171&idb=0>.

Дополнительная литература:

1. Булгаков Д. А. Программирование интерактивных приложений на языке C# : учеб. пособие / Булгаков Д. А. - Санкт-Петербург : ГУАП, 2022. - 159 с. - Книга из коллекции ГУАП - Информатика. - ISBN 978-5-8088-1776-0., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=867106&idb=0>.

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

<https://gitlab.manytask.org/cpp0-nn/public-2024-fall>

7. Материально-техническое обеспечение дисциплины (модуля)

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

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 01.04.02 - Прикладная математика и информатика.

Автор(ы): Золотых Николай Юрьевич, доктор физико-математических наук, доцент.

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

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