

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 президиума Ученого совета ННГУ

протокол от

«30» ноября 2022 г. № 13

Рабочая программа дисциплины

Программирование на C++. Часть 1

(наименование дисциплины (модуля))

Уровень высшего образования

магистратура

(бакалавриат / магистратура / специалитет)

Направление подготовки / специальность

02.04.02 Фундаментальная информатика и информационные технологии

(указывается код и наименование направления подготовки / специальности)

Направленность образовательной программы

Искусственный интеллект

(указывается профиль / магистерская программа / специализация)

Форма обучения

очная

(очная / очно-заочная / заочная)

Нижний Новгород

2023 год

1. Место дисциплины в структуре ООП

Дисциплина Б1.В.ДВ.03.02 «Программирование на C++. Часть1» относится к дисциплинам по выбору части, формируемой участниками образовательных отношений, Блока 1 «Дисциплины, модули» направления 02.04.02 «Фундаментальная информатика и информационные технологии», направленность «Искусственный интеллект», преподается в 1 семестре.

№ Варианта	Место дисциплины в учебном плане образовательной программы	Стандартный текст для автоматического заполнения в конструкторе РПД
1	Блок 1. Дисциплины (модули) Часть, формируемая участниками образовательных отношений. Дисциплина по выбору	Дисциплина Б1.В.ДВ.03.02 «Программирование на C++. Часть1» относится к части ООП направления подготовки 02.04.02 Фундаментальная информатика и информационные технологии, формируемой участниками образовательных отношений.

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-8. Способен к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности	ПК-8.1. Знать основы ИТ и иметь навыки анализа современного состояния науки и информационных технологий в этой области	Знает основы ИТ в области КС и КГ и имеет навыки анализа современного состояния науки и информационных технологий в этой области	собеседование,
	ПК-8.2. Иметь навыки проектирования и разработки и развития ИТ-решений на основе анализа современного состояния науки и информационных технологий	Имеет навыки проектирования и разработки и развития ИТ-решений на основе анализа современного состояния науки и информационных технологий в области КС и КГ	задания

3. Структура и содержание дисциплины

3.1. Трудоемкость дисциплины

	очная форма обучения
--	-----------------------------

Общая трудоемкость	3 ЗЕТ
Часов по учебному плану	108
в том числе	
контактная работа:	33
- занятия лекционного типа	32
- занятия семинарского типа	32
- текущий контроль (КСР)	1
самостоятельная работа	43
Промежуточная аттестация – зачет	

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				Саостоятельная работа
		Занятия лекционного типа	Занятия семинарского типа	Занятия лабораторного типа	Всего	
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
Текущий контроль (КСР)	1				1	
Промежуточная аттестация –зачет						
Итого	108	32	32		65	43

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

Практическая подготовка предусматривает: Разработка приложений: с использованием семантики перемещения; с использованием функциональных объектов и лямбда-выражений; разработке параллельных программ с использованием C++ потоков; разработка программ с использованием примитивов синхронизации C++; реализация шаблонных контейнеров с количеством элементов, известных на этапе компиляции; освоение типовых техник метапрограммирования.

На проведение практических занятий (семинарских занятий) в форме практической подготовки отводится 16 часов.

Практическая подготовка направлена на формирование и развитие:

- практических навыков в соответствии с профилем ОП: Разработка, тестирование, оптимизация программного обеспечения (ПО). Разработка технической документации на продукцию в сфере ИТ.
- компетенций – ПК-8: Способен к разработке новых алгоритмических, методических и технологических решений в конкретной сфере профессиональной деятельности (ПК-8.2: Иметь навыки проектирования и разработки и развития ИТ-решений на основе анализа современного состояния науки и информационных технологий).

Текущий контроль успеваемости реализуется в формах опросов на занятиях семинарского типа Промежуточная аттестация проходит в традиционной форме зачета. Зачет проставляется после набора заданного числа баллов.

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Для выполнения программы самостоятельной работы дисциплины «Программирование на C++. Часть1» 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` --- 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>

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, известных 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 компетен-	Шкала оценивания сформированности компетенций						
	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	Не зачтено		Зачтено				

ций)		
<u>Знания</u>	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающ	<div data-bbox="513 280 668 788">Уровень знаний ниже минимальных требований. Имели место грубые ошибки.</div> <div data-bbox="668 280 833 788">Минимально допустимый уровень знаний. Допущено много негрубых ошибки.</div> <div data-bbox="833 280 997 788">Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок</div> <div data-bbox="997 280 1161 788">Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок</div> <div data-bbox="1161 280 1326 788">Уровень знаний в объеме, соответствующем программе подготовки, без ошибок.</div> <div data-bbox="1326 280 1495 788">Уровень знаний в объеме, превышающем программу подготовки.</div>

	егося от ответа						
<u>Умения</u>	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	При решении стандартных задач 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 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 Примеры контрольных вопросов для проверки ПК-8

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

14) Реализация <code>std::tuple</code> с использованием различных стандартов языка C++.	ПК-8
---	------

5.2.2. Типовые задания для оценки сформированности компетенции ПК-8

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 <code>[from, to)</code> 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 <code>Split(string, delimiter)</code> , которая разбивает строку <code>string</code> на подстроки по заданному разделителю <code>delimiter</code> . Например, <code>Split("just a test", " ") = {"just", "a", "test"}</code> . Обработать и разделители идущие подряд.
6.	Реализуйте функцию <code>EvaluateExpression</code> , которая вычисляет целочисленное выражение в обратной польской записи (операции <code>+</code> , <code>*</code> , <code>-</code> , целые числа разделенные пробелами).
60.	Реализуйте функцию <code>CreateYandexForecaster</code> , которая возвращает текущую погоду, используя API Яндекс.Погоды. Данная функция принимает ключ для доступа к API и <code>url</code> , к которому нужно делать запросы (используется в тестах). Интерфейсы приведены в файле <code>weather.h</code> , реализуйте все необходимое в <code>weather.cpp</code> . Реализация интерфейса <code>IForecaster</code> позволяет принимать ширину-долготу, которые нужно использовать при запросе.
61.	Написать telegram-бота, используя библиотеку <code>POCO</code> . 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 ресурсе: <https://gitlab.com/moskalenkoviktor>

6. Учебно-методическое и 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 С++».
<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, оснащенные оборудованием и техническими средствами обучения: операционная система Windows (лицензия), Microsoft Visual Studio (лицензия).

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 02.04.02 «Фундаментальная информатика и информационные технологии».

Автор Золотых Н.Ю.

Заведующий кафедрой Золотых Н.Ю.

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