

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 2**

(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.07.02 «Программирование на C++. Часть2» 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 во 2 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.07.02 «Программирование на C++. Часть 2» 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-10. Способен конвертировать результаты научно- исследовательских и/или опытно-конструкторских работ в требования ИТ-проекта, и обратно: способен обеспечить ИТ-проект необходимым исследованием и опытно-конструкторскими работами.	ПК-10.1. Знать проблематику и методы научных исследований и опытно-конструкторских ИТ-разработок с применением C++.	<b>Знать</b> теоретические основы, алгоритмы и постановки задач компьютерной графики и глубокого обучения и средства их профессионального программирования на C++.	собеседование
	ПК-10.2. Иметь навыки выполнения научных исследований и опытно-конструкторских ИТ-разработок с применением C++	<b>Владеть</b> навыками программирования на C++ для создания системных решений в области машинного и глубокого обучения	собеседование, задания

## 3. Структура и содержание дисциплины

### 3.1. Трудоемкость дисциплины

	<b>очная форма обучения</b>
<b>Общая трудоемкость</b>	<b>3 ЗЕТ</b>
<b>Часов по учебному плану</b>	<b>108</b>

<b>в том числе</b>	
<b>контактная работа:</b>	<b>33</b>
- занятия лекционного типа	16
- занятия семинарского типа	16
- текущий контроль (КСР)	1
<b>самостоятельная работа</b>	<b>75</b>
<b>Промежуточная аттестация – зачет</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			Всего	Самостоятельная работа
		Занятия лекционного типа	Занятия семинарского типа	Занятия лабораторного типа		
1. Потоки. Мьютексы, guard-ы, data race. RW-lock, дедлоки, TSAN.	7	1	1		2	5
2. Условная синхронизация, condition variable.	9	1	1		2	7
3. Модель памяти и атомарные переменные. Барьеры.	13	2	2		4	9
4. Реализация примитивов синхронизации на linux. Futex. Hazard pointer.	13	2	2		4	9
5. Метапрограммирование шаблонов. SFINAE. constexpr. Концепты.	13	2	2		4	9
6. Сетевое программирование.	13	2	2		4	9
7. Алгоритмы работы санитайзеров (asan, msan, tsan).	13	2	2		4	9
8. Архитектура компилятора, clang, llvm.	13	2	2		4	9
9. Производительность. Устройство RAM. Микроархитектура процессора. Perf.	13	2	2		4	9
Текущий контроль (КСР)	1				1	
Промежуточная аттестация –зачет						
<b>Итого</b>	<b>108</b>	<b>16</b>	<b>16</b>		<b>33</b>	<b>75</b>

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: создание программ на языке C++ со следующими объектами: контейнеры STL; итераторы; тип данных auto; умные указатели; шаблоны проектирования с использованием умных указателей; функциональные объекты; лямбда-выражения.

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-10: Способен конвертировать результаты научно- исследовательских и/или опытно-конструкторских работ в требования ИТ-проекта, и обратно: способен обеспечить ИТ-проект необходимым исследованием и опытно-конструкторскими работами. (ПК-10.2: Иметь навыки применения результатов научно- исследовательских и опытно-конструкторских работ в ИТ-проектах в области КС.).

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++. Часть2» достаточно: (а) самостоятельной проработки лекционного и дополнительного материала и (б) выполнить практическую часть дисциплины.

Практическая часть курса предлагает к решению 37 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 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 обучающ	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.

	егося от ответа						
<u>Умения</u>	Отсутств ие минимал ьных умений. Невозмо жность оценить наличие умений вследств ие отказа обучающ егося от ответа	При решении стандартны х задач не продемонст рованы основные умения.  Имели место грубые ошибки.	Продемонст рованы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме.	Продемонст рованы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продемонст рованы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продемонст рованы все основные умения, решены все основные задачи с отдельными несущест венным недочетами, выполнены все задания в полном объеме.	Продемонст рованы все основные умения, решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
<u>Навыки</u>	Отсутств ие владения материал ом. Невозмо жность оценить наличие навыков вследств ие отказа обучающ егося от ответа	При решении стандартны х задач не продемонст рованы базов ые навык и.  Имели место грубые ошибки.	Имеется минимальны й набор навыков для решения стандартных задач с некоторыми недочетами.	Продемонст рованы базовые навыки при решении стандартных задач с некоторыми недочетами	Продемонст рованы базовые навыки при решении стандартных задач без ошибок и недочетов.	Продемонст рованы навыки при решении нестандартн ых задач без ошибок и недочетов.	Продемонст рованы творческий подход к решению нестандартн ых 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-10

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

13) Параллельные программы на C++. Подход к разработке.	ПК-10
14) Реализация <code>std::tuple</code> с использованием различных стандартов языка C++.	ПК-10

### 5.2.2. Типовые задания для оценки сформированности компетенции ПК-10

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

Примеры содержания задач	
1.	Реализовать многопоточную версию функции, проверяющей число на простоту <code>bool is_prime(uint64_t)</code> - простой поиск делителей до корня числа. Исследуйте время работы однопоточной и многопоточной версии. Реализовать алгоритм так, чтобы другие потоки не совершали холостой работы, если какой-то поток уже нашел делитель числа.
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 поддерживающую многопоточность хеш-таблицу (Concurrent Hash Map), поддерживающую отображение ключ -> значение. Базовая реализация находится в файле <code>concurrent_hash_map.h</code> и использует <code>std::unordered_map</code> и мьютекс для блокирования перед всеми операциями, что неэффективно при наличии большого числа потоков, одновременно работающих с таблицей.
27.	Реализовать декодер <code>jpg</code> . Обязательная часть - реализация <code>baseline sequential</code> режима. <code>Progressive</code> режим - 500 бонусных баллов. Установить библиотеки: <code>sudo apt-get install libjpeg-dev libpng-dev libfftw3-dev</code> . Для IDCT использовать <a href="http://www.fftw.org/">http://www.fftw.org/</a> ; декодирование Хаффмана вынести в отдельный класс; декодер оформить в виде класса, принимающего <code>std::istream&amp;</code> .
28.	<code>clang-fun</code> – написать утилиту с использованием <code>clang tooling</code> ( <a href="https://clang.llvm.org/docs/LibTooling.html">https://clang.llvm.org/docs/LibTooling.html</a> ) для проверки соответствия имен стилю Google Styleguide ( <a href="https://google.github.io/styleguide/cppguide.html#Naming">https://google.github.io/styleguide/cppguide.html#Naming</a> ), а также искать опечатки в этих именах. Проверять только имена переменных, членов классов, типов, констант, функций и перечислений. Ваша утилита также должна удовлетворять этому стилю.

Полный список задач см. на ресурсе: <https://gitlab.com/moskalenkoviktor/shad-cpp-nn>

## 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 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, 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.