

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 президиума Ученого совета ННГУ

протокол № 1 от 16.01.2024 г.

Рабочая программа дисциплины

Программирование на C++. Часть 2

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

2024 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.07.02 Программирование на C++. Часть 2 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: Знать основы ИТ в области КС и иметь навыки анализа современного состояния науки и информационных технологий в области КС ПК-8.2: Иметь навыки проектирования и разработки и развития ИТ-решений на основе анализа современного состояния науки и информационных технологий в области КС ПК-8.3: Иметь навыки управления разработкой и развитием ИТ-решений на основе анализа современного состояния науки и информационных технологий в области КС	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	Практическое задание	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	

аудиторные занятия (контактная работа):	
- занятия лекционного типа	16
- занятия семинарского типа (практические занятия / лабораторные работы)	16
- КСР	1
самостоятельная работа	75
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/ лабора торные работы), часы	Всего	
	0 ф 0	0 ф 0	0 ф 0	0 ф 0	0 ф 0
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
Аттестация	0				
КСР	1				1
Итого	108	16	16	33	75

Содержание разделов и тем дисциплины

1. Потоки. Мьютексы, guard-ы, data race. RW-lock, дедлоки, TSAN.
2. Условная синхронизация, condition variable.
3. Модель памяти и атомарные переменные. Барьеры.
4. Реализация примитивов синхронизации на linux. Futex. Hazard pointer.
5. Метапрограммирование шаблонов. SFINAE. constexpr. Концепты.
6. Сетевое программирование.
7. Алгоритмы работы санитайзеров (asan, msan, tsan).
8. Архитектура компилятора, clang, llvm.
9. Производительность. Устройство RAM. Микроархитектура процессора. Perf.

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.

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>

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 - Практическое задание) для оценки сформированности компетенции ПК-8:

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>

Примеры задач

1. Реализовать многопоточную версию функции, проверяющей число на простоту `bool is_prime(uint64_t)` - 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 $\text{BinPow}(a,b,c)$, которая вычисляет a

$b \pmod{c}$. Количество операций должно быть пропорционально двоичному логарифму b .

3. Реализовать поддерживающую многопоточность хеш-таблицу (Concurrent HashMap), поддерживающую отображение ключ -> значение. Базовая реализация находится в файле `concurrent_hash_map.h` и использует `std::unordered_map` и мьютекс для блокирования перед всеми операциями, что неэффективно при наличии большого числа потоков, одновременно работающих с таблицей.

4. Реализовать декодер `jpg`. Обязательная часть - реализация `baseline sequential` режима. `Progressive` режим - 500 бонусных баллов. Установить библиотеки: `sudo apt-get install`

`libjpeg-dev libpng-dev libfftw3-dev`. Для IDCT использовать <http://www.fftw.org/>;

декодирование Хаффмана вынести в отдельный класс; декодер оформить в виде класса, принимающего `std::istream&`.

5. `clang-fun` – написать утилиту с использованием `clang tooling`

(<https://clang.llvm.org/docs/LibTooling.html>) для проверки соответствия имен стилю `Google Styleguide`

(<https://google.github.io/styleguide/cppguide.html#Naming>), 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>

Критерии оценивания (оценочное средство - Практическое задание)

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 (индикатор)	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	не зачтено		зачтено				

достижения							
<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 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 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	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 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 - 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 13.12.2023, протокол № 3.