

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

Язык C++

Уровень высшего образования

Магистратура

Направление подготовки / специальность

01.04.02 - Прикладная математика и информатика

Направленность образовательной программы

Компьютерные науки и приложения

Форма обучения

очная

г. Нижний Новгород

2024 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.01.01 Язык C++ 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-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. ПК-12.2: Умеет оценивать трудоемкость разработки программных средств для решения задач производственно-технологической деятельности ПК-12.3: Имеет навыки разработки системного программного обеспечения для решения задач производственно-технологической деятельности.	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 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 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	Задания	Зачёт: Контрольные вопросы
ПК-5: Способен	ПК-5.1: Знает типовые	ПК-5.1:	Задания	

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: 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/лабораторные работы), часы	Всего	
	0 ф 0	0 ф 0	0 ф 0	0 ф 0	0 ф 0
Rvalue-ссылки и семантика перемещений	23	4	4	8	15
Лямбда-выражения.	28	4	4	8	20
Параллельные вычисления.	28	4	4	8	20
Метапрограммирование	28	4	4	8	20
Аттестация	0				
КСР	1				1
Итого	108	16	16	33	75

Содержание разделов и тем дисциплины

Функции, переменные, типы. Операторы и выражения. Компиляция, заголовочные файлы. Указатели и ссылки. Параметры и возвращаемые значения функций. Арифметика указателей. Структуры и классы. ООП, конструкторы, деструкторы, операторы. Инкапсуляция. Шаблоны и STL. Контейнеры, алгоритмы, итераторы. Функторы и лямбды. Динамическое управление памятью. Умные указатели. RAII. Наследование. Виртуальные функции, абстрактные классы. Исключения. Паттерны проектирования. Расположение объектов в памяти, выравнивание. Отладка, GDB, core dumps. Строки. Кодировки и регулярные выражения. Move-семантика. rvalue-ссылки, 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 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>

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

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 - Задания) для оценки сформированности компетенции ПК-12:

1. 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.

2. Написать telegram-бота, используя библиотеку РОСО. В задании будет оцениваться дизайн и реализация клиента для общения с серверами telegram. Реализация 3 дополнительных функций – до 400 премиальных баллов

3. 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>

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

1. Задача для знакомства с тестирующей системой. Реализовать функцию multiply, которая перемножает 2 числа, в файле multiplication.h
2. Реализуйте функцию быстрого возведения в степень BinPow(a, b, c), которая вычисляет $a^b \pmod c$. Количество операций должно быть пропорционально двоичному логарифму b.
3. Реализовать функцию Range(from, to, step), аналог функции range из python2 - возвращает вектор всех чисел из диапазона [from, to) с шагом step. Примеры в тест-коде.

4. Реализовать функцию Unique, 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).

Критерии оценивания (оценочное средство - Задания)

Оценка	Критерии оценивания
зачтено	Все практические задания выполнены в полном объеме и в срок. Описание всех этапов выполнения заданий, код и результаты работы представлены преподавателю
не зачтено	Выполнены не все практические задания или выполнены не в полном объеме (представлено не полное описание этапов выполнения заданий, код работает некорректно, результаты работы не представлены преподавателю).

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатор 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	Продemonстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но	Продemonстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания	Продemonстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но	Продemonстрированы все основные умения. Решены все основные задачи с отдельным и несуществ	Продemonстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов

			не в полном объеме	в полном объеме, но некоторые с недочетами	некоторые с недочетами	енными недочетам и, выполнены все задания в полном объеме	
<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 некоторым и недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторым и 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 - Контрольные вопросы) для оценки сформированности компетенции ПК-12

1) Реализация семантики перемещения в контейнерах STL.

2) Методы по умолчанию, генерируемые в классах.
3) Списки инициализации. Плюсы и минусы.
4) Функциональные объекты, функции, лямбда-выражения, функторы. Плюсы и минусы.
5) Является ли функция объявленная с модификатором const потокобезопасной? Примеры.
6) Параллельные программы на C++. Подход к разработке.
7) Реализация std::tuple с использованием различных стандартов языка C++.

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

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.

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
зачтено	Владение основным и дополнительным материалом достаточное или с незначительными ошибками и погрешностями
не зачтено	владение материалом, необходимым по данному предмету, недостаточно. Работу за время семестра можно оценить как неудовлетворительную

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

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 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 13.12.2023, протокол № 3.