

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 27.08.2025

Рабочая программа дисциплины

Профессиональный C++

Уровень высшего образования
Бакалавриат

Направление подготовки / специальность
02.03.02 - Фундаментальная информатика и информационные технологии

Направленность образовательной программы
Системное программирование

Форма обучения
очная

г. Нижний Новгород

2025 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.01.01 Профессиональный 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
ПК-4: Способен проектировать программное обеспечение	<p>ПК-4.3: Осуществляет обоснованный выбор методов и алгоритмов для программной реализации формальной математической модели</p> <p>ПК-4.4: Разрабатывает программный код с использованием языков программирования</p> <p>ПК-4.5: Оформляет программный код в соответствии с установленными требованиями</p> <p>ПК-4.6: Работает с системой управления версиями программного кода</p> <p>ПК-4.7: Проверяет и отлаживает программный код</p>	<p>ПК-4.3:</p> <p>ПК-4.3. 3-1. Знает основные классы методов программной реализации моделей и критерии выбора алгоритмов</p> <p>ПК-4.3. У-1. Умеет проводить сравнительный анализ и обоснование выбора алгоритмов для программной реализации модели</p> <p>ПК-4.3. В-1. Владеет методами адаптации методов и алгоритмов под специфику задачи</p> <p>ПК-4.3. В-2. Владеет навыками оценки эффективности выбранных алгоритмов</p> <p>ПК-4.4:</p> <p>ПК-4.4. 3-1. Знает синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования</p> <p>ПК-4.4. 3-2. Знает методологии разработки компьютерного программного обеспечения</p> <p>ПК-4.4. 3-3. Знает технологии программирования</p> <p>ПК-4.4. У-1. Умеет применять выбранные языки</p>	Задания	Зачёт: Контрольные вопросы

		<p>программирования для написания программного кода</p> <p>ПК-4.4. У-2. Умеет использовать выбранную среду программирования.</p> <p>ПК-4.4. У-3. Умеет использовать возможности имеющейся технической и/или программной архитектуры для написания программного кода</p> <p>ПК-4.5:</p> <p>ПК-4.5. З-1. Знает нормативно-технические документы (стандарты и регламенты), определяющие требования к оформлению программного кода</p> <p>ПК-4.5. З-2. Знает основные стандарты оформления технической документации на компьютерное программное обеспечение</p> <p>ПК-4.5. У-1. Умеет применять заданные стандарты и шаблоны для составления и оформления технической документации</p> <p>ПК-4.5. У-2. Умеет применять нормативно-технические документы (стандарты и регламенты), определяющие требования к оформлению программного кода</p> <p>ПК-4.5. У-3. Умеет применять инструментарий для создания и актуализации исходных текстов программ</p> <p>ПК-4.6:</p> <p>ПК-4.6. З-1. Знает возможности используемой системы управления версиями и вспомогательных инструментальных программных средств</p> <p>ПК-4.6. З-2. Знает установленный регламент использования системы управления версиями</p> <p>ПК-4.6. У-1. Умеет</p>		
--	--	--	--	--

		<p>регистривать изменения исходного текста программного кода в системе управления версиями</p> <p>ПК-4.6. У-2. Умеет сохранять изменения программного кода в соответствии с регламентом управления версиями</p> <p>ПК-4.6. У-3. Умеет выполнять слияние, разделение и сравнение исходных текстов программного кода</p> <p>ПК-4.7:</p> <p>ПК-4.7. 3-1. Знает методы и приемы отладки программного кода</p> <p>ПК-4.7. 3-2. Знает типы и форматы сообщений об ошибках, предупреждений</p> <p>ПК-4.7. 3-3. Знает способы использования технологических журналов, форматы и типы записей журналов</p> <p>ПК-4.7. У-1. Умеет выявлять ошибки в программном коде</p> <p>ПК-4.7. У-2. Умеет отлаживать программный код на уровне программных модулей</p> <p>ПК-4.7. У-3. Умеет отлаживать программный код на уровне межмодульных взаимодействий и взаимодействий с окружением</p>		
<p>ППК-Р6: Способен участвовать в промышленной разработке программного обеспечения (top)</p>	<p>ППК-Р6.1: Работает в соответствии с промышленными методологиями разработки</p> <p>ППК-Р6.2: Использует инструменты промышленной разработки</p> <p>ППК-Р6.3: Разрабатывает масштабируемый и поддерживаемый код</p>	<p>ППК-Р6.1:</p> <p>ППК-Р6.1. 3-1. Знает принципы Agile и их применение в промышленных проектах.</p> <p>ППК-Р6.1. 3-2. Знает процессы code review, принципы коллективного владения кодом (collective code ownership).</p> <p>ППК-Р6.1. У-1. Умеет оценивать объем задачи и срок ее выполнения, участвовать в планировании спринтов.</p>	<p>Задания</p>	<p>Зачёт: Контрольные вопросы</p>

		<p>ППК-Р6.1. У-2. Умеет работать в команде с использованием инструментов управления проектами.</p> <p>ППК-Р6.2:</p> <p>ППК-Р6.2. 3-1. Знает принципы Continuous Integration and Continuous Delivery (CI/CD).</p> <p>ППК-Р6.2. 3-2. Знает системы мониторинга и логирования в продуктивной среде.</p> <p>ППК-Р6.2. У-1. Умеет настраивать потоки работ CI/CD.</p> <p>ППК-Р6.2. У-2. Умеет работать с контейнеризацией и оркестрацией.</p> <p>ППК-Р6.2. У-2. Умеет настраивать мониторинг в продуктивной среде.</p> <p>ППК-Р6.3:</p> <p>ППК-Р6.3. 3-1. Знает принципы чистого кода, SOLID, DRY, KISS и др.</p> <p>ППК-Р6.3. 3-2. Знает принципы предметно-ориентированного проектирования (ПОП) программного обеспечения.</p> <p>ППК-Р6.3. 3-3. Знает паттерны проектирования и антипаттерны.</p> <p>ППК-Р6.3. У-1. Умеет разрабатывать модульный и тестируемый программный код.</p> <p>ППК-Р6.3. У-2. Умеет выполнять модульное, интеграционное и нагрузочное тестирование.</p> <p>ППК-Р6.3. У-3. Умеет проводить рефакторинг для повышения качества кода.</p> <p>ППК-Р6.3. У-4. Умеет применять принципы ПОП при разработке программного обеспечения на языках программирования высокого уровня абстракций и в</p>		
--	--	--	--	--

		LowCode u NoCode системах		
--	--	---------------------------	--	--

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очная
Общая трудоемкость, з.е.	2
Часов по учебному плану	72
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	16
- занятия семинарского типа (практические занятия / лабораторные работы)	16
- КСР	1
самостоятельная работа	39
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			Самостоятельная работа обучающегося, часы
		Контактная работа (работа во взаимодействии с преподавателем), часы из них			
		Занятия лекционного типа	Занятия семинарского типа (практические занятия/лабораторные работы), часы	Всего	
Ф	Ф	Ф	Ф	Ф	
Введение. Состав языка программирования C++. Эволюция C++.	9	2	2	4	5
Базовые концепции языка C++. Объявления и определения. Правило одного определения.	9	2	2	4	5
Строки в языке C++. С-строки, строки в виде класса. Подходы к повышению эффективности строк	9	2	2	4	5
Шаблоны в C++. Шаблоны функций, вывод типов, шаблоны и перегрузка. Шаблоны классов. Двухфазное разрешение имен в шаблонах.	9	2	2	4	5
Механизм исключений в C++. Гарантии безопасности. Проектирование кода с учетом исключений	9	2	2	4	5
Левые и правые ссылки в C++. Семантика перемещений	9	2	2	4	5
Лямбда-выражения в C++. Замыкание и захват в лямбда-выражении	9	2	2	4	5
Интеллектуальные указатели в C++. Идиома RAII. Уникальное и совместное владение ресурсами.	8	2	2	4	4
Аттестация	0				

КСР	1			1	
Итого	72	16	16	33	39

Содержание разделов и тем дисциплины

Цель дисциплины:

Целью освоения дисциплины «Профессиональный С++» является изучение расширенных возможностей современных стандартов языка программирования С++.

Изучаются типовые проблемы, возникающие в рамках индустриального программирования на С++, и методы их решения. Основное внимание уделяется развитию навыков использования современных языковых конструкций, введенных в стандартах С++ 11, С++ 14, С++ 17.

Целью освоения дисциплины «Профессиональный С++» является изучение расширенных возможностей современных стандартов языка программирования С++.

Изучаются типовые проблемы, возникающие в рамках индустриального программирования на С++, и методы их решения. Основное внимание уделяется развитию навыков использования современных языковых конструкций, введенных в стандартах С++ 11, С++ 14, С++ 17.

Целью освоения дисциплины «Профессиональный С++» является изучение расширенных возможностей современных стандартов языка программирования С++.

Изучаются типовые проблемы, возникающие в рамках индустриального программирования на С++, и методы их решения. Основное внимание уделяется развитию навыков использования современных языковых конструкций, введенных в стандартах С++ 11, С++ 14, С++ 17.

Целью освоения дисциплины «Профессиональный С++» является изучение расширенных возможностей современных стандартов языка программирования С++.

Изучаются типовые проблемы, возникающие в рамках индустриального программирования на С++, и методы их решения. Основное внимание уделяется развитию навыков использования современных языковых конструкций, введенных в стандартах С++ 11, С++ 14, С++ 17.

Целью освоения дисциплины «Профессиональный С++» является изучение расширенных возможностей современных стандартов языка программирования С++.

Изучаются типовые проблемы, возникающие в рамках индустриального программирования на С++, и методы их решения. Основное внимание уделяется развитию навыков использования современных языковых конструкций, введенных в стандартах С++ 11, С++14, С++ 17, С++20.

Содержание дисциплины:

1. Введение. Состав языка программирования С++. Эволюция С++.
2. Базовые концепции языка С++. Объявления и определения. Правило одного определения.
3. Строки в языке С++. С-строки, строки в виде класса. Подходы к повышению эффективности строк.
4. Шаблоны в С++. Шаблоны функций, вывод типов, шаблоны и перегрузка. Шаблоны классов. Двухфазное разрешение имен в шаблонах.
5. Механизм исключений в С++. Гарантии безопасности. Проектирование кода с учетом исключений.
6. Левые и правые ссылки в С++. Семантика перемещений.
7. Лямбда-выражения в С++. Замыкание и захват в лямбда-выражении.
8. Интеллектуальные указатели в С++. Идиома RAII. Уникальное и совместное владение ресурсами.

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.

1. Вjarne Stroustrup. A Tour of C++, 3rd Edition. – Addison-Wesley Professional, 2022.
2. Бьерн Страуструп. Язык программирования C++. Специальное издание. – М.: Бином, 2011.
3. Николай Джосаттис: Стандартная библиотека C++: справочное руководство, 2-е изд. – М.: Вильямс, 2014. – 1136 с.
4. Герб Саттер. Новые сложные задачи на C++. Серия "C++ In-Depth". – М.: Вильямс, 2015. – 272 с.
5. Герб Саттер. Решение сложных задач на C++. Серия "C++ In-Depth" – М.: Вильямс, 2015. – 400 с.
6. Мейерс С. Эффективный и современный C++: 42 рекомендации по использованию C++11 и C++14. – М.: Вильямс, 2016.

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-4:

1. Освоение использования контейнеров STL на типовых задачах.
2. Освоение работы с итераторами как с основой доступа к элементам контейнера.

5.1.2 Типовые задания (оценочное средство - Задания) для оценки сформированности компетенции ППК-Р6:

1. Освоение механизма вывода типов C++ и лучших практик его использования.
2. Освоение использования интеллектуальных указателей и типовых вариантов их применения.

Критерии оценивания (оценочное средство - Задания)

Оценка	Критерии оценивания
зачтено	Выполнены все или большая часть этапов решения задачи или задача решена с незначительными недочетами. Код и результаты работы представлены преподавателю в срок.
не зачтено	Выполнены не все лабораторные работы 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 (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 - Контрольные вопросы) для оценки сформированности компетенции ПК-4

1. Почему невозможно копировать объект типа `std::unique_ptr`? Примеры.
2. Что такое `std::weak_ptr`? Зачем он нужен? Примеры.
3. Типичные шаблоны проектирования с использованием интеллектуальных указателей.

5.3.2 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ППК-Р6

1. Типичные шаблоны проектирования с использованием интеллектуальных указателей.
2. Оптимизация классов с использованием семантики перемещения.
3. Реализация семантики перемещения в контейнерах STL.
4. Методы по умолчанию, генерируемые в классах.
5. Списки инициализации. Плюсы и минусы.
6. Функциональные объекты, функции, лямбда-выражения, функторы. Плюсы и минусы.
7. Является ли функция объявленная с модификатором `const` потокобезопасной? Примеры.

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
зачтено	Студент ответил на большую часть вопросов возможно с незначительными недочетами.
не зачтено	При ответе студент допускает грубые ошибки в основном материале и решении стандартных задач.

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

Основная литература:

1. Язык программирования С++ для профессионалов / Страуструп Б. - Москва : ИНТУИТ, 2016. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=663690&idb=0>.

Дополнительная литература:

1. МейерсС. Наиболее эффективное использование С++. 35 новых рекомендаций по улучшению ваших программ и проектов : учебно-методическое пособие / МейерсС. - Москва : ДМК-пресс, 2016. - 29 с. - ISBN 978-5-97060-193-8. - Текст : электронный., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=868696&idb=0>.

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

Интернет-ресурсы:

<https://e-lib.unn.ru/>

<https://intuit.ru/>

Программное обеспечение:

1. Операционные системы семейства Microsoft Windows, лицензия по подписке Microsoft Imagine.
2. Браузер Google Chrome, предоставляется бесплатно на условиях лицензионных соглашений на программное обеспечение с открытым исходным кодом.
3. 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 02.03.02 - Фундаментальная информатика и информационные технологии.

Автор(ы): Сысоев Александр Владимирович, кандидат технических наук, доцент.

Заведующий кафедрой: Мееров Иосиф Борисович, кандидат технических наук.

Программа одобрена на заседании методической комиссии от 25.06.2025, протокол № Протокол №11.