

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
президиумом Ученого совета ННГУ
от 14.12.2021 г протокол № 4

Рабочая программа дисциплины

Профессиональный C++

(наименование дисциплины (модуля))

Уровень высшего образования

Бакалавриат

(бакалавриат / магистратура / специалитет)

Направление подготовки / специальность

09.03.04 Программная инженерия

(указывается код и наименование направления подготовки / специальности)

Направленность образовательной программы

Разработка программно-информационных систем

(указывается профиль / магистерская программа / специализация)

Форма обучения

очная

(очная / очно-заочная / заочная)

Нижний Новгород

2022 год

1. Место дисциплины в структуре ОПОП

Дисциплина относится к факультативам.

№ варианта	Место дисциплины в учебном плане образовательной программы	Стандартный текст для автоматического заполнения в конструкторе РПД
3	ФТД. Факультативы	Дисциплина <i>ФТД.02 Профессиональный</i> C++ является факультативом в ООП направления подготовки 09.03.04. Программная инженерия

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	
<i>ПК-12. Способен осуществлять концептуальное, функциональное и логическое проектирование систем и подсистем малого и среднего масштаба и сложности.</i>	<i>ПК-12.1 Знает методы концептуального, функционального и логического проектирования</i>	<i>Знать</i> <i>Назначение и особенности использования основных модулей и компонент стандартов C++ 98, C++ 11, C++ 14, C++ 17.</i> <i>Знать</i> <i>Область применения основных модулей и компонент стандартов C++ 98, C++ 11, C++ 14, C++ 17.</i>	<i>Собеседование</i>
	<i>ПК-12.3. Умеет применять архитектурные стили и паттерны проектирования при решении типовых задач</i>	<i>Уметь</i> <i>Эффективно применять на практике основные компоненты стандартов C++ 98, C++ 11, C++ 14, C++ 17.</i>	<i>Задача</i>

	<i>ПК-12.4. Владеет навыками разработки проекта программной системы с учетом возможностей и ограничений</i>	<i>Пользоваться навыками разработки и применять их в зависимости от особенностей задачи только необходимых компонент стандартов C++ 98, C++ 11, C++ 14, C++ 17.</i>	<i>Задача</i>

3. Структура и содержание дисциплины

3.1. Трудоемкость дисциплины

	очная форма обучения
Общая трудоемкость	1 ЗЕТ
Часов по учебному плану	36
в том числе	
контактная работа:	33
- занятия лекционного типа	16
-занятия семинарского типа	16
- текущий контроль (КСР)	1
самостоятельная работа	3
Промежуточная аттестация – зачет	

3.2. Содержание дисциплины

Наименование и краткое содержание 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	Занятия лабораторного типа	Всего	
Введение. Основные элементы стандарта C++ 98. Переход к современному C++.	4	4			4	
Вывод типов.	4	2	2		4	
Интеллектуальные указатели.	4	2	2		4	
Rvalue-ссылки и семантика перемещений.	4	2	2		4	

Лямбда-выражения.	4	2	2		4	
Параллельные вычисления.	7	2	2		4	3
Метапрограммирование.	8	4	4		8	
Текущий контроль (КСР)	1				1	
Промежуточная аттестация –зачет						
Итого	36	16	16		33	3

Текущий контроль успеваемости реализуется в формах опросов на занятиях семинарского типа.

Промежуточная аттестация проходит в традиционной форме (зачет).

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Для выполнения программы самостоятельной работы дисциплины «Профессиональный C++» достаточно: (а) самостоятельной проработки лекционного и дополнительного материала и (б) выполнить по согласованию с преподавателем 7 лабораторных работ на темы, представленные ниже в таблице. Лабораторные работы выполняются в среде Microsoft Visual Studio.

№ п / п	Раздел дисциплины	Наименование лабораторных работ
1	Введение. Основные элементы стандарта C++ 98. Переход к современному C++.	1) Освоение использования контейнеров STL на типовых задачах их использования. 2) Освоение работы с итераторами как с основой доступа к элементам контейнера.
2	Вывод типов.	1) Освоение механизма вывода типов C++ и лучших практик его использования.
3	Интеллектуальные указатели.	1) Основы использования интеллектуальных указателей и типовых шаблонов их применения.
4	Rvalue-ссылки и семантика перемещений.	1) Освоение использования семантики перемещения.
5	Лямбда-выражения.	1) Освоение использования функциональных объектов и лямбда-выражений.
6	Параллельные вычисления.	1) Освоение подходов к разработке параллельных программ с использованием потоков C++. 2) Освоение лучших практик использования примитивов синхронизации C++.
7	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 (индикатора 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 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 негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки, без ошибок.	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 несущественным недочетами, выполнены все задания в полном объеме.	Продemonстрированы все основные умения, решены все основные задачи. Выполнены все задания

							ния, в пол ном объ еме без нед очет ов
<u>Навыки</u>	Отсутствие владения материалом. 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.	Продemonстрированы 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 Контрольные вопросы

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

5.2.2. Типовые задачи для оценки сформированности компетенции ПК-12

- 1) Контейнеры STL и типовые задачи их использования.
- 2) Итераторы как основа доступа к элементам контейнеров STL.
- 3) Механизм вывода типов C++, лучшие практики его использования.
- 4) Интеллектуальные указатели и типовые шаблоны их применения.
- 5) Использование семантики перемещения.
- 6) Функциональные объекты и лямбда-выражения.
- 7) Подходы к разработке параллельных программ с использованием потоков C++.
- 8) Лучшие практики использования примитивов синхронизации C++.
- 9) Шаблонные контейнеры с количеством элементов, которое известно на этапе компиляции.

10) Типовые техники метапрограммирования.

5.2.3. Вопросы для собеседования

- 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++.

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:

3. Огнева, М. В. Программирование на языке c++: практический курс : учебное пособие для бакалавриата и специалитета / М. В. Огнева, Е. В. Кудрина. — М. : Издательство Юрайт, 2017. — 335 с. — (Серия : Бакалавр и специалист). — ISBN 978-5-534-05123-0. — Режим доступа : www.biblio-online.ru/book/7670D7EC-AC37-4675-8EAE-DD671BC6D0E4.

1. Фридман А. «Язык программирования C++».
<http://www.intuit.ru/studies/courses/17/17/info>

в) программное обеспечение и Интернет-ресурсы:

1. Единое окно доступа к образовательным ресурсам (Программирование)
<http://window.edu.ru>
2. Электронно-библиотечная система <https://e.lanbook.com/>
3. <https://msdn.microsoft.com/ru-ru/dn308572.aspx>

7. Материально-техническое обеспечение дисциплины

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой (лекционного и семинарского типа), оснащенные оборудованием и 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 09.03.04 Программная инженерия.

Авторы

_____ Н.А. Борисов

_____ А.А. Сиднев

Рецензент (ы) _____

Заведующий кафедрой _____ Р.Г. Стронгин

Программа одобрена на заседании методической комиссии института информационных технологий, математики и механики

от 1.12.2021 года, протокол № 2.