

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Московский физико-технический институт  
(национальный исследовательский университет)»**

**УТВЕРЖДЕНО**  
**Проректор по учебной работе**

**А.А. Воронов**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Современный C++
<b>по направлению:</b>	Прикладные математика и физика
<b>профиль подготовки:</b>	Радиотехника и компьютерные технологии Физтех-школа Радиотехники и Компьютерных Технологий кафедра проблем передачи информации и анализа данных
<b>курс:</b>	4
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 8 (весенний) - Зачет

Аудиторных часов: 30 всего, в том числе:

лекции: 30 час.

семинары: 0 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 15 час.

Всего часов: 45, всего зач. ед.: 1

Программу составил: А.А. Крещук, канд. техн. наук

Программа обсуждена на заседании кафедры проблем передачи информации и анализа данных 20.04.2020

## Аннотация

Современные программы на языке программирования C++ значительно отличаются от тех, что были написаны десять лет назад. Отличия обусловлены как изменениями в самом языке и его стандартной библиотеке, так и появлением новых библиотек и вспомогательных программ. Данный курс научит студентов разрабатывать программы, соответствующие современным стандартам качества.

### 1. Цели и задачи

#### Цель дисциплины

Научить студентов программировать на C++ с учётом сложившихся практик написания безопасного и эффективного программного кода.

#### Задачи дисциплины

- ознакомить студентов с наиболее распространёнными ошибками, приводящими к «undefined behavior» и «ill-formed, no diagnostic required»;
- обучить студентов использовать современные средства языка C++, позволяющих писать более простой код и упрощающих поиск ошибок;
- познакомить студентов с наиболее важными вспомогательными программами и библиотеками;
- оказание консультаций студентам в проведении собственных теоретических и экспериментальных исследований, включающих в себя разработку программ на языке программирования C++.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1 Анализирует задачу, выделяя этапы ее решения, действия по решению задачи
	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки
	УК-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности
	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации
ПК-4 Способен критически оценивать применимость используемых методик и методов	ПК-4.1 Знает численные порядки величин, характерных для соответствующей профессиональной области
	ПК-4.3 Способен обосновать причинно-следственные отношения используемых понятий и моделей

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- основные понятия стандарта языка программирования C++;
- основные классы и функции стандартной библиотеки C++.

уметь:

- проектировать архитектуру программы, упрощающую задачу написания безопасного и эффективного кода;
- локализовывать ошибки и bottleneck в программе.

владеть:

- навыком освоения большого объёма информации;
- навыком быстрого чтения исходного текста программ и библиотек.

#### 4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

##### 4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Основные вспомогательные программы.	4			2
2	Возможности языка C++.	10			5
3	Стандартная библиотека.	10			5
4	Основные сторонние библиотеки.	6			3
Итого часов		30			15
Подготовка к экзамену		0 час.			
Общая трудоёмкость		45 час., 1 зач.ед.			

##### 4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 8 (Весенний)

###### 1. Основные вспомогательные программы.

Системы сборки: CMake, meson. Форматирование кода: clang-format. Статический анализ: clang-tidy, cppcheck. Отладка: gdb, lldb, их графические оболочки, интеграция со средами разработки. Профилирование: valgrind, perf. Benchmarking: google-benchmark. Веб-сайты: godbolt.org, cppinsights.io.

###### 2. Возможности языка C++.

Система типов. Шаблоны: шаблоны переменных, автоматический вывод аргументов шаблона, SFINAE, шаблонное метапрограммирование, концепты. Вычисления времени компиляции.

Пространства имён: простые, анонимные, inline.

Функциональное программирование. Лямбда-выражения.

Исключения и noexcept. Коды ошибки. Контракты.

Обратная совместимость, feature test macros.

###### 3. Стандартная библиотека.

Современные коллекции. `std::ranges`, концепты итераторов, `range adaptors`. Алгоритмы.  
Работа с файловой системой.  
Работа с памятью: `std::unique_ptr` и `std::shared_ptr`. Аллокатеры. `Polymorphic memory resource`.  
Параллельное программирование. Потoki, мьютексы, `condition_variable`. Атомарные операции.  
`Promise` и `future`. Параллельные алгоритмы, отличия между `std::execution::unsequenced_policy` и `std::execution::parallel_policy`.  
Важные `type-erasure`: `std::function`, `std::span`, `std::string_view`.

#### 4. Основные сторонние библиотеки.

`Boost.Hana` — современное метапрограммирование на функциях.  
`Eigen` — линейная алгебра.  
`Pybind11` — интерфейс для `python`.

### 5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Учебная аудитория, оснащенная мультимедийным оборудованием (проектор или плазменная панель), доской. Индивидуальные компьютеры или ноутбуки.

### 6. Перечень рекомендуемой литературы

#### Основная литература

Фонд литературы базовой кафедры (организации):

1. Stroustrup B. The C++ Programming Language // (4th Edition). Addison-Wesley, May 2013. ISBN 978-0321563842.
2. Stroustrup B. A Tour of C++ // (Second edition). Addison-Wesley, July 2018. ISBN 978-0-13-499783-4.

#### Дополнительная литература

### 7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

<https://cppreference.com>  
<https://godbolt.org>  
<https://cppinsights.io/>

### 8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

На лекционных занятиях используются мультимедийные технологии, включая демонстрацию презентаций.

### 9. Методические указания для обучающихся по освоению дисциплины (модуля)

Студент, изучающий дисциплину, должен, с одной стороны, овладеть общими понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике. В результате изучения дисциплины студент должен знать основные понятия стандарта языка программирования C++, основные классы и функции стандартной библиотеки, основные шаблоны архитектуры программ.

Успешное освоение курса требует напряженной самостоятельной работы студента. В программе курса отведено минимально необходимое время для работы студента над темой. Самостоятельная работа включает в себя:

- чтение и конспектирование рекомендованной литературы;

- проработку учебного материала (по конспектам занятий, учебной и научной литературе), подготовку ответов на вопросы, предназначенные для самостоятельного изучения, решение задач;

- подготовка к зачёту.

Руководство и контроль за самостоятельной работой студента осуществляется в форме индивидуальных консультаций.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов следует обращаться за консультациями к лектору.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Прикладные математика и физика
<b>профиль подготовки:</b>	Радиотехника и компьютерные технологии Физтех-школа Радиотехники и Компьютерных Технологий кафедра проблем передачи информации и анализа данных
<b>курс:</b>	4
<b>квалификация:</b>	бакалавр
Семестр, формы промежуточной аттестации: 8 (весенний) - Зачет	
<b>Разработчик:</b>	А.А. Крещук, канд. техн. наук

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1 Анализирует задачу, выделяя этапы ее решения, действия по решению задачи
	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.3 Рассматривает различные варианты решения задачи, оценивает их преимущества и недостатки
	УК-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук, и использовать их в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности
	ОПК-4.4 Владеет навыками работы с компьютером и компьютерными сетями с целью получения, хранения и обработки научной (технической, технологической) информации
ПК-4 Способен критически оценивать применимость используемых методик и методов	ПК-4.1 Знает численные порядки величин, характерных для соответствующей профессиональной области
	ПК-4.3 Способен обосновать причинно-следственные отношения используемых понятий и моделей

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Современный C++» обучающийся должен:

### знать:

- основные понятия стандарта языка программирования C++;
- основные классы и функции стандартной библиотеки C++.

### уметь:

- проектировать архитектуру программы, упрощающую задачу написания безопасного и эффективного кода;
- локализовывать ошибки и bottleneck в программе.

### владеть:

- навыком освоения большого объема информации;
- навыком быстрого чтения исходного текста программ и библиотек.

## 3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

С целью контроля освоения обучающимися учебного материала проводится устный опрос в начале занятия по теме прошлой лекции или в конце занятия по пройденной теме.

## 4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

Перечень контрольных вопросов для сдачи зачета:

1. сборка и работа под Windows и Linux,
2. обработка всех ошибок и валидация входных данных,
3. многопоточная реализация,
4. интерфейсы для C++ и python,
5. анализ производительности.

#### Критерии оценивания

Зачет - выставляется студенту, показавшему владение основными положениями курса, умеет использовать полученные знания при решении типовых практических задач.

Незачет - выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

#### **5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Зачет может проводиться по итогам текущей успеваемости и путем организации специального опроса, проводимого в устной и (или) письменной форме.

При проведении зачета обучающемуся предоставляется 20 минут на подготовку. Опрос обучающегося проводится в течение 30 минут.

Во время проведения зачета обучающиеся могут пользоваться программой дисциплины.