

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Ливанов Дмитрий Викторович  
Должность: Ректор  
Дата подписания: 19.03.2026 10:36:59  
Уникальный программный ключ:  
с6d909c49c1d2034fa3a0156c4eaa51e7230c3a1

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

## ВЫПИСКА ИЗ ПРОТОКОЛА № 12

заседания учебно-методического совета от 12 июля 2023

ПОВЕСТКА:

Рассмотрение дополнительных общеобразовательных и профессиональных программ.

Проректор по учебной работе А. А. Воронов.

СЛУШАЛИ: Директора Физтех-школы прикладной математики и информатики  
А.М. Райгородского.

ПОСТАНОВИЛИ:

Рекомендовать к утверждению в установленном порядке дополнительную  
общеобразовательную программу «Разработчик C++ Junior».

Решение принято единогласно заочным голосованием.

Форма проведения заседания: заочная.

Председатель УМС МФТИ

А.А. Воронов

Ученый секретарь УМС  
МФТИ

М.В. Березникова

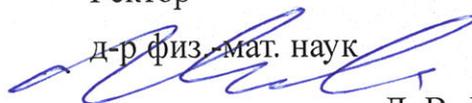
Министерство науки и высшего образования Российской Федерации  
федеральное государственное автономное образовательное учреждение высшего  
образования

«Московский физико-технический институт  
(национальный исследовательский университет)»

УТВЕРЖДАЮ

Ректор

д-р физ.-мат. наук



Д. В. Ливанов

«12» июля 2023 г.

**Дополнительная общеобразовательная программа**

**«Разработчик C++ Junior»**

Москва 2023

## **Оглавление**

1. Общая характеристика программы	2
2. Планируемые результаты обучения	3
3. Учебный план	3
4. Учебная программа	4
5. Организационно-педагогические условия реализации дополнительной профессиональной программы	16
5.1. Требования к материально-техническим условиям	16
5.2. Требования к информационным и учебно-методическим условиям	17
6. Оценка качества освоения программы	19
7. Составители программы:	19

## 1. Общая характеристика программы

- 1.1. Целью реализации дополнительной общеобразовательной программы «Разработчик C++ Junior» (далее – программа) является получение навыков разработки программного обеспечения на языке C++, на технологиях, популярных в среде разработчиков C++. Будут рассмотрены основы программирования на C++, средства разработки, сборки и тестирования программ, основные алгоритмы и структуры данных, современные инструменты разработки программного обеспечения.
- 1.2. Категории слушателей, на обучение которых рассчитана программа: слушатели с отсутствующим опытом программирования и создания приложений на языке программирования C++. Участники получают основные сведения об объектно-ориентированном программировании, терминологии и синтаксисе языка, научатся создавать простые программы на языке программирования C++, изучат алгоритмы и структуры данных, инструменты Linux, Git, Docker, технологии и инструменты CI/CD.
- 1.3. Нормативный срок освоения программы – 620 академических часов.
- 1.4. Форма обучения: очная (с применением дистанционных образовательных технологий).
- 1.5. Режим обучения: дистанционно, 14 ак. часов в неделю.

## 2. Планируемые результаты обучения

В результате освоения программы слушатель должен:

владеть навыками:

- написание эффективного и безопасного кода на C++;
- работа с операционной системой Linux;
- работа с системой контроля версий GIT;

знать:

- базовые понятия программирования;
- область применения и синтаксис C++;
- принципы написания эффективного и безопасного кода на C++;
- основные концепции ООП;
- основные алгоритмы и структуры данных;

уметь:

- писать эффективный и безопасный код на C++;
- работать с операционной системой Linux;
- работать с системой контроля версий GIT;
- работать с инструментом контейнеризации Docker;
- работать с инструментами CI/CD.

## 3. Учебный план

№ п/п*	Наименование Модулей, дисциплин	Всего, час.	В том числе:				
			Лекции	Практические занятия (семинары)	Самостоятельная работа	Контрольные задания	Форма контроля
1	Модуль 1. Основы программирования	60	12	12	30	6	проект
2	Модуль 2. Разработка на C++	400	100	100	176	24	проект
3	Модуль 3. Linux / Git для начинающих	20	7	6	5	2	тестирование
4	Модуль 4. Введение в алгоритмы и структуры данных	80	16	16	40	8	тестирование
5	Модуль 5. Docker и CI/CD для начинающих	20	2	1	10	7	тестирование
6	Итоговая квалификационная работа	40			40		проект

	<b>ИТОГО:</b>	<b>620</b>	<b>137</b>	<b>135</b>	<b>301</b>	<b>47</b>	
--	---------------	------------	------------	------------	------------	-----------	--

\* Слушатели могут проходить модули в произвольном порядке.

#### 4. Учебная программа

Учебно-тематический план программы

##### Модуль 1. Основы программирования

№ п/п	Содержание обучения, наименование и тематика практических занятий (вебинаров), самостоятельных работы	Объем, ак.час.
<b>1.1.</b>	<b>Введение. Условные операторы</b>	<b>10</b>
	Теория: <ol style="list-style-type: none"> <li>1. Компилируемые/интерпретируемые языка программирования.</li> <li>2. Среды разработки, онлайн компиляторы, jupyter - notebook.</li> <li>3. Ввод / вывод информации.</li> <li>4. Типы данных и арифметические действия.</li> <li>5. Переменная и ее место в памяти. Стек и куча. Имя переменной, значение переменной.</li> <li>6. Условный оператор, решение задач. Блок if-elif-else. Операции сравнения, in.</li> </ol>	2
	Практика: установка среды разработки, решение задач на условные операторы.	2
	Самостоятельная работа.	5
	Выполнение контрольных заданий.	1
<b>1.2.</b>	<b>Вложенные условия. Циклы while / for</b>	<b>10</b>
	Теория: <ol style="list-style-type: none"> <li>1. Логические операции or, and, not</li> <li>2. Вложенные условия if, равносильность условию elif</li> <li>3. Цикл с предусловием и постусловием.</li> <li>4. Операции break, continue.</li> <li>5. Цикл for, зачем нужен цикл счетчик.</li> </ol>	2

	Практика: решения сложных задач, знакомство с трассировкой и отладкой, решение задач на ввод чисел вплоть до конкретного условия, решение задач на тему вложенных циклов.	2
	Самостоятельная работа.	5
	Выполнение контрольных заданий.	1
<b>1.3.</b>	<b>Введение в коллекции. Вложенные списки</b>	<b>10</b>
	Теория: <ol style="list-style-type: none"> <li>1. Списки. Операции со списками и со строками. Индексация списков</li> <li>2. Хранение данных, ввод данных.</li> <li>3. Список списков. Хранение данных в памяти компьютера.</li> <li>4. Срезки строк и списков.</li> </ol>	2
	Практика: решение задач на списки.	2
	Самостоятельная работа.	5
	Выполнение контрольных заданий.	1
<b>1.4.</b>	<b>Функции</b>	<b>10</b>
	Теория: <ol style="list-style-type: none"> <li>1. Понятие функции и функционального программирования. Принимаемое константное число аргументов функции.</li> <li>2. Область видимости переменных. Копирование или неkopирование элементов. Возвращаемые значения.</li> <li>3. Вызов функции в памяти компьютера. Переменное число аргументов.</li> <li>4. Лямбда функции.</li> </ol>	2
	Практика: решение задач на функции.	2
	Самостоятельная работа.	5
	Выполнение контрольных заданий.	1
<b>1.5.</b>	<b>Введение в ООП</b>	<b>10</b>
	Теория: <ol style="list-style-type: none"> <li>1. Понятие объектно-ориентированного программирования. Что такое класс, экземпляр и метод класса.</li> </ol>	2

	<ul style="list-style-type: none"> <li>2. Основные принципы ООП.</li> <li>3. Понятие полиморфизма. Определение и переопределение операторов.</li> <li>4. Инкапсуляция.</li> <li>5. Отличие Python от других языков программирования.</li> </ul>	
	Практика: реализация ООП в Python.	2
	Самостоятельная работа.	5
	Выполнение контрольных заданий.	1
<b>1.6.</b>	<b>Наследование. Итоговый проект</b>	<b>10</b>
	Теория: <ul style="list-style-type: none"> <li>1. Наследование.</li> <li>2. Отличия наследования в Python от других языков программирования.</li> <li>3. Проектирование и разработка продукта, методика.</li> <li>4. Что такое Python, и почему он объектно ориентирован. В чем различие с другими языками программирования.</li> </ul>	2
	Практика: реализация написание итогового проекта	2
	Самостоятельная работа.	5
	Выполнение контрольных заданий.	1
	<b>Итого</b>	<b>60</b>

## Модуль 2. Разработка на C++

№ п/п	Содержание обучения, наименование и тематика практических занятий (вебинаров), самостоятельных работы	Объем, ак.час.
<b>2.1</b>	<b>Модуль 1. Основы C++</b>	<b>127</b>
2.1.1.	Занятие 1. Введение в информатику.	4,5
2.1.2.	Занятие 2. Установка средств разработки. Работа со средой.	5,5

2.1.3.	Занятие 3. Введение в язык C++: состав языка, структура программы.	8
2.1.4	Занятие 4. Типы данных: концепция типа данных, примитивные типы языка C++, допустимые операции.	7
2.1.5	Занятие 5. Переменные и выражения. Классы памяти. Операция приведения типов.	7
2.1.6	Занятие 6. Обсуждение последних занятий, домашних работ. Работа над проектом.	8
2.1.7	Занятие 7. Управляющие операторы: ветвление (if, switch).	7
2.1.8	Занятие 8. Управляющие операторы: циклы (for, while, do-while).	7
2.1.9	Занятие 9. Структурное программирование. Функции. Рекурсия. Перегрузка функций.	8
2.1.10	Занятие 10. Обсуждение последних занятий, домашних работ. Работа над проектом.	7
2.1.11	Занятие 11. Указатели и ссылки. Передача и возврат значений по указателю/ссылке.	7
2.1.12	Занятие 12. Массивы: статические массивы, динамические массивы, связь массивов и указателей.	8
2.1.13	Занятие 13. C-style строки.	7
2.1.14	Занятие 14. Обсуждение последних занятий, домашних работ. Работа над проектом.	7
2.1.15	Занятие 15. Модульное программирование: заголовочные файлы, файлы реализации.	8
2.1.16	Занятие 16. Обобщенное программирование. Шаблоны функций. Стандартная библиотека алгоритмов.	7

2.1.17	Занятие 17. Перечисления, структуры. Шаблоны структур. Списки.	7
2.1.18	Занятие 18. Обсуждение последних занятий, домашних работ. Работа над проектом.	8
<b>2.2</b>	<b>Модуль 2. Основы объектно-ориентированного программирования</b>	<b>117</b>
2.2.1.	Занятие 1. Введение в ООП. Классы и методы. Модификаторы доступа.	7
2.2.2.	Занятие 2. Константные и статические поля и методы. Друзья класса.	7
2.2.3.	Занятие 3. Конструкторы. Перегрузка операций.	8
2.2.4	Занятие 4. Обсуждение последних занятий, домашних работ. Работа над проектом.	7
2.2.5	Занятие 5. Деструкторы. Управление ресурсами. RAII.	7
2.2.6	Занятие 6. Семантика перемещения-I.	8
2.2.7	Занятие 7. Семантика перемещения-II.	7
2.2.8	Занятие 8. Обсуждение последних занятий, домашних работ. Работа над проектом.	7
2.2.9	Занятие 9. Умные указатели-I.	8
2.2.10	Занятие 10. Умные указатели-II.	7
2.2.11	Занятие 11. Последовательные контейнеры.	7
2.2.12	Занятие 12. Обсуждение последних занятий, домашних работ. Работа над проектом.	8
2.2.13	Занятие 13. Итераторы.	7

2.2.14	Занятие 14. Ассоциативные контейнеры.	7
2.2.15	Занятие 15. Функторы. Лямбда-выражения.	8
2.2.16	Занятие 16. Обсуждение последних занятий, домашних работ. Работа над проектом.	7
<b>2.3</b>	<b>Модуль 3. Наследование и обработка ошибок</b>	<b>117</b>
2.3.1.	Занятие 1. Наследование-I.	7
2.3.2.	Занятие 2. Наследование-II.	7
2.3.3.	Занятие 3. Множественное наследование.	8
2.3.4	Занятие 4. Обсуждение последних занятий, домашних работ. Работа над проектом.	7
2.3.5	Занятие 5. Динамический полиморфизм-I.	7
2.3.6	Занятие 6. Динамический полиморфизм-II.	8
2.3.7	Занятие 7. Динамический полиморфизм-III.	7
2.3.8	Занятие 8. Обсуждение последних занятий, домашних работ. Работа над проектом.	7
2.3.9	Занятие 9. Исключения-I.	8
2.3.10	Занятие 10. Исключения-II.	7
2.3.11	Занятие 11. Безопасный код. Гарантии безопасности исключений.	7
2.3.12	Занятие 12. Обсуждение последних занятий, домашних работ. Работа над проектом.	8

2.3.13	Занятие 13. Ручное управление памятью и временем жизни объектов.	7
2.3.14	Занятие 14. Вариабельные шаблоны.	7
2.3.15	Занятие 15. Обсуждение последних занятий, домашних работ. Работа над проектом.	8
2.3.16	Занятие 16. Обсуждение последних занятий, домашних работ. Работа над проектом.	7
<b>2.4</b>	<b>Итоговая аттестация</b>	<b>39</b>
	<b>Итого</b>	<b>400</b>

### Модуль 3. Linux / Git для начинающих

№ п/п	Содержание обучения, наименование и тематика практических занятий (вебинаров), самостоятельных работы	Объем, ак.час.
<b>3.1</b>	<b>Базовый WSL и VirtualBox</b>	<b>3,5</b>
	Теория: <ul style="list-style-type: none"> <li>1. Обзор курса. План занятий</li> <li>2. Введение. Чем отличается Windows и Linux</li> <li>3. Установка WSL. Запуск</li> <li>4. VirtualBox установка и использование. Тест</li> </ul>	1
	Практическая работа по теме лекции	1
	Самостоятельная работа	1
	Выполнение контрольных заданий	0,5
<b>3.2</b>	<b>Основы Linux</b>	<b>6,5</b>
	Теория: <ul style="list-style-type: none"> <li>1. Введение. Какие задачи он решает. Дистрибутив Ubuntu</li> <li>2. Консоль. Основа работы с Linux. tmux</li> <li>3. Скрипты. Продвинутая работа с консолью</li> </ul>	2,5

	<ul style="list-style-type: none"> <li>4. Файловая система</li> <li>5. Работа с сетью</li> <li>6. Серверное взаимодействие. SSH</li> </ul>	
	Практическая работа по теме лекции	2,5
	Самостоятельная работа	1
	Выполнение контрольных заданий	0,5
<b>3.3</b>	<b>WSL подробно</b>	<b>4</b>
	Теория: <ul style="list-style-type: none"> <li>1. Краткий экскурс в историю появления. WSL1 и WSL 2</li> <li>2. Основные команды работы с WSL</li> <li>3. Файловая система</li> <li>4. Сеть WSL</li> <li>5. WSL USB</li> <li>6. CUDA и Windows Докер</li> </ul>	1,5
	Практическая работа по теме лекции	1
	Самостоятельная работа	1
	Выполнение контрольных заданий	0,5
<b>3.4</b>	<b>Git</b>	<b>6</b>
	Теория: <ul style="list-style-type: none"> <li>1. Введение. Цели. Разновидности систем контроля версий</li> <li>2. Базовые основы GIT. Популярные сценарии работы</li> <li>3. Серверные репозитории Git. Ветки. Merge request</li> <li>4. Сценарии отката изменений. Stash</li> </ul>	2
	Практическая работа по теме лекции	1,5
	Самостоятельная работа	2
	Выполнение контрольных заданий	0,5
	<b>Итого</b>	<b>20</b>

#### Модуль 4. Введение в алгоритмы и структуры данных

№ п/п	Содержание обучения, наименование и тематика практических занятий (вебинаров), самостоятельных работы	Объем, ак.час.
<b>4.1.</b>	<b>Введение</b>	<b>10</b>
	Теория: <ol style="list-style-type: none"> <li>1. Орг. вопросы, Яндекс-контеcт, работа в системе, O-семантика, асимптотические классы, Мастер-теорема.</li> <li>2. Обзор контейнеров в разных языках программирования, основные операции и их асимптотика.</li> <li>3. Массивы, списки, двусвязные списки, стек и очередь.</li> </ol>	2
	Практика: Реализация некоторых контейнеров. Правильные скобочные последовательности. Задача о книжках. Обратная польская нотация. Обработка запросов к серверу.	2
	Самостоятельная работа: Написать свой стек, список (двусвязанный и односвязный), очередь на стеке. Решение задач на очереди/стеки.	5
	Выполнение контрольных заданий.	1
<b>4.2.</b>	<b>Сортировки</b>	<b>10</b>
	Теория: <ol style="list-style-type: none"> <li>1. Понятие, примеры использования и свойства сортировок.</li> <li>2. Квадратичные сортировки – сортировки вставками и пузырьком.</li> <li>3. Merge Sort.</li> </ol>	2
	Практика: Разбор дз. Семинар по Merge Sort / Quick Sort.	2
	Самостоятельная работа: Написать обе квадратичные сортировки, merge sort (quicksort - опционально).	5
	Выполнение контрольных заданий.	1
<b>4.3.</b>	<b>Динамическое программирование. Бинарный (экспоненциальный) поиск.</b>	<b>10</b>
	Теория: <ol style="list-style-type: none"> <li>1. Задача поиска числа в контейнере. Вычисление асимптотики. Бинарный поиск - зачем использовать.</li> <li>2. Бинарный и экспоненциальный поиск.</li> </ol>	2

	<p>3. Динамическое программирование - что это и зачем надо. Задача о рюкзаке. (Не)точность дп.</p> <p>4. Задача о рюкзаке с дополнительным условием. Двумерное динамическое программирование.</p>	
	Практика: Бинарный и экспоненциальный поиск - реализация.	2
	Самостоятельная работа: Решение задач на бинарный поиск. Динамическое программирование одномерное и двумерное.	5
	Выполнение контрольных заданий.	1
<b>4.4.</b>	<b>Хеширование. Метод цепочек.</b>	<b>10</b>
	<p>Теория:</p> <ol style="list-style-type: none"> <li>1. Хеш-функция. Ее свойства, реализация для различных типов данных. Проблема коллизии.</li> <li>2. Хеш-таблица. Контейнеры в различных языках, использующие хеш-таблицу. Алгоритм работы хеш-таблицы.</li> <li>3. ООП, атрибуты класса хеш-таблицы. Реализация хеш-таблицы методом закрытой адресации. Расчет ее асимптотик.</li> </ol>	2
	Практика: Ответы на вопросы, разбор домашнего задания.	2
	Самостоятельная работа: Задачи на хеш функцию. Построение своей хеш-таблицы с закрытой адресацией.	5
	Выполнение контрольных заданий.	1
<b>4.5.</b>	<b>Хеш-таблицы с открытой адресацией</b>	<b>10</b>
	<p>Теория:</p> <ol style="list-style-type: none"> <li>1. Разрешение коллизии методом двойного хеширования.</li> <li>2. Две хеш-функции, их свойства. Реализация алгоритма.</li> </ol>	2
	Практика: Обсуждение домашнего задания, его разбор. Решение задач по теме: хеш-таблицы / ответы на вопросы.	2
	Самостоятельная работа: Решить задачу для открытой адресации. Описать различия избежания коллизий.	5
	Выполнение контрольных заданий.	1
<b>4.6.</b>	<b>Куча</b>	<b>10</b>
	<p>Теория:</p> <ol style="list-style-type: none"> <li>1. Определение и свойства кучи.</li> </ol>	2

	<p>2. Вставка, Поиск максимума, Удаление. Реализация абстрактного типа данных.</p> <p>3. Эффективное построение кучи. Взятие максимума/минимума. Какие задачи решает - в чем эффективность кучи.</p> <p>4. Пирамидальная сортировка.</p>	
	Практика: Построение и реализация кучи.	2
	Самостоятельная работа: Построение кучи, взятие максимума/минимума, пирамидальная сортировка.	5
	Выполнение контрольных заданий.	1
<b>4.7.</b>	<b>Простые деревья и граф</b>	<b>10</b>
	<p>Теория:</p> <ol style="list-style-type: none"> <li>1. Определение дерева. Высота дерева, родительские, дочерние узлы. Способы хранения дерева.</li> <li>2. Обход в глубину и ширину.</li> <li>3. Поиск ключа. Вставка/удаление. Проблема несбалансированности дерева. Оценка Эффективности.</li> <li>4. Решаемые задачи. Поиск компонент связности, проверка на двудольность. Поиск цикла.</li> </ol>	2
	Практика: Реализация обхода. Решение задач по теме.	2
	Самостоятельная работа: DFS, BFS, решение задач с урока.	5
	Выполнение контрольных заданий.	1
<b>4.8.</b>	<b>Сбалансированные деревья</b>	<b>10</b>
	<p>Теория:</p> <ol style="list-style-type: none"> <li>1. Напоминание: зачем балансировать деревья. Мотивация использовать графы. Определение AVL-дерева.</li> <li>2. AVL - дерево.</li> <li>3. Вращения дерева. Операции вставки и удаления.</li> <li>4. Оценка эффективности AVL-дерева.</li> </ol>	2
	Практика: Реализация конструкции класса. Реализация алгоритма AVL-дерева и ответы на вопросы.	2
	Самостоятельная работа: Реализация AVL-дерева.	5
	Выполнение контрольных заданий.	1

	<b>Итого</b>	<b>80</b>
--	--------------	-----------

### Модуль 5. Docker и CI/CD для начинающих

№ п/п	Содержание обучения, наименование и тематика практических занятий (вебинаров), самостоятельных работы	Объем, ак.час.
<b>5.1</b>	<b>Docker</b>	<b>6,5</b>
	Теория: <ol style="list-style-type: none"> <li>1. Введение</li> <li>2. Что такое докер? Как его используют в организациях? Лицензирование и аналоги</li> <li>3. Принципы работы докер. Докер vs VM</li> <li>4. Контейнер vs образ. Dockerfile</li> <li>5. Работа с контейнером. docker exec</li> <li>6. Создание внешних дисков, подключение директорий</li> <li>7. Создание сетей. Типы сетей</li> <li>8. Docker Best Practices</li> </ol>	1
	Практическая работа по теме лекции	0,5
	Самостоятельная работа	3
	Выполнение контрольных заданий	2
<b>5.2</b>	<b>Docker Compose</b>	<b>5,5</b>
	Теория: <ol style="list-style-type: none"> <li>1. Что такое Docker Compose? Примеры и преимущества</li> <li>2. Docker Compose. Сеть и диски в docker compose</li> </ol>	0,5
	Практическая работа по теме лекции	0
	Самостоятельная работа	3
	Выполнение контрольных заданий	2
<b>5.3</b>	<b>DevOps. GitLab CI</b>	<b>8</b>

	Теория: 1. Что такое DevOps? Какая связь с Agile, Scrum? Введение пайплайны в CI/CD 2. GitLab CI. Структура пайплайнов 3. CI/CD в GitLab CI 4. Подведение итогов	0,5
	Практическая работа по теме лекции	0,5
	Самостоятельная работа	4
	Выполнение контрольных заданий	3
	<b>Итого</b>	<b>20</b>

<b>6.</b>	<b>Итоговая проектная работа</b>	<b>40</b>
	Самостоятельная работа.	40
	<b>Итого</b>	<b>620</b>

## 5. Организационно-педагогические условия реализации дополнительной профессиональной программы

### 5.1. Требования к материально-техническим условиям

Перечень кабинетов, лабораторий, мастерских, тренажеров и др., обеспечивающих проведение всех предусмотренных программой видов занятий:

Наименование специализированных аудиторий, кабинетов, лабораторий	Вид занятий	Наименование оборудования, программного обеспечения
Система дистанционного обучения провайдера массовых открытых онлайн курсов	Лекции	Слушателю необходимо наличие доступа в сеть интернет, компьютер.  Преподавателю курса необходимо наличие доступа администратора курса на LMS-платформе к материалам курса.

Информационно-коммуникационная платформа дистанционных семинаров	Практические занятия (дистанционные семинары)	Слушателю необходимо наличие доступа в сеть интернет, компьютер.  Преподавателю курса необходимо оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.
Система дистанционного обучения провайдера массовых открытых онлайн курсов	Самостоятельная работа	Наличие компьютера и доступа в сеть интернет.
Система дистанционного обучения провайдера массовых открытых онлайн курсов	Промежуточный, рубежный контроль, Итоговая аттестация	Наличие компьютера и доступа в сеть интернет.

## 5.2. Требования к информационным и учебно-методическим условиям

Список литературы:

### МОДУЛЬ 1:

Основная литература

1. Никитина, Т. П. Программирование. Основы Python для инженеров / Т. П. Никитина, Л. В. Королев. — Санкт-Петербург : Лань, 2023. — 156 с. — ISBN 978-5-507-45284-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/302720>. — Режим доступа: для авториз. пользователей.

Дополнительная литература

1. Задачи по программированию. Под ред. С. М. Окулова, - М.: БИНОМ. Лаборатория знаний, 2016.
2. Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. Алгоритмы: построение и анализ, 3-е издание. — М.: Вильямс, 2013.
3. Соколов А.П. Системы программирования: теория, методы, алгоритмы. - М.:ФиС, 2014.
4. С. М. Окулов. Основы программирования. - М.: Бином. Лаборатория знаний, 2012.
5. Совершенный код. Мастер-класс/ Пер. с англ. — М. : Издательство «Русская редакция», 2010. — 896 стр.

### МОДУЛЬ 2:

Основная литература

1. Страуструп, Б. Язык программирования C++ для профессионалов : учебное пособие / Б. Страуструп. — 2-е изд. — Москва : ИНТУИТ, 2016. — 670 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100542>

(дата обращения: 31.03.2023). — Режим доступа: для авториз. пользователей.

#### Дополнительная литература

1. Стивен Прата. Язык программирования C++. Лекции и упражнения.
2. Роберт Лафоре. Объектно-ориентированное программирование.
3. Скотт Мейерс. Эффективный современный C++.

### МОДУЛЬ 3:

#### Основная литература

1. Орещенков, И. С. Операционные системы. Bodhi Linux 6.0: установка, настройка, эксплуатация / И. С. Орещенков. — Санкт-Петербург : Лань, 2023. — 364 с. — ISBN 978-5-507-44988-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/276659> (дата обращения: 29.03.2023). — Режим доступа: для авториз. пользователей.

#### Дополнительная литература

1. «Операционные системы. Программное обеспечение : учебник / составитель Т. П. Куль. — Санкт-Петербург : Лань, 2020. — 248 с. — ISBN 978-5-8114-4290-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131045> (дата обращения: 29.03.2023). — Режим доступа: для авториз. пользователей.
2. Донцов, В. П. Linux на примерах : руководство / В. П. Донцов, И. В. Сафин. — Санкт-Петербург : Наука и Техника, 2017. — 352 с. — ISBN 978-5-94387-742-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/101550> (дата обращения: 29.03.2023). — Режим доступа: для авториз. пользователей.
3. Фишерман, Л. В. Git. Практическое руководство. Управление и контроль версий в разработке программного обеспечения : руководство / Л. В. Фишерман. — Санкт-Петербург : Наука и Техника, 2021. — 304 с. — ISBN 978-5-94387-547-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/191470> (дата обращения: 29.03.2023). — Режим доступа: для авториз. пользователей.

### МОДУЛЬ 4:

#### Основная литература

1. Седжвик, Р. Алгоритмы на C++ : учебное пособие / Р. Седжвик. — 2-е изд. — Москва : ИНТУИТ, 2016. — 1772 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100565> (дата обращения: 29.03.2023). — Режим доступа: для авториз. пользователей.

#### Дополнительная литература

1. Бабенко, М. А. Введение в теорию алгоритмов и структур данных / М. А. Бабенко, М. В. Левин. — Москва : МЦНМО, 2016. — 144 с. — ISBN 978-5-4439-2396-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/80136> (дата обращения: 29.03.2023). — Режим доступа: для авториз. пользователей.
2. Хайнеман Д., Поллис Г., Селков С. Алгоритмы. Справочник с примерами на C, C++, Java и Python. — 2017.

## МОДУЛЬ 5:

### Основная литература

1. Сейерс, Э. Х. Docker на практике / Э. Х. Сейерс, А. Милл ; перевод с английского Д. А. Беликов. — Москва : ДМК Пресс, 2020. — 516 с. — ISBN 978-5-97060-772-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131719> (дата обращения: 29.03.2023). — Режим доступа: для авториз. пользователей.

### Дополнительная литература

1. Моуэт, Э. Использование Docker / Э. Моуэт ; научный редактор А. А. Маркелов ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2017. — 354 с. — ISBN 978-5-97060-426-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/93576> (дата обращения: 29.03.2023). — Режим доступа: для авториз. пользователей.

### 6. Оценка качества освоения программы

Оценка качества освоения программы осуществляется в процессе промежуточной аттестации. Формы и методы промежуточного контроля представлены в таблице.

Наименование модулей	Основные показатели оценки	Форма аттестации
Основы программирования	Качество выполнения проектной работы	проект
Разработка на C++	Качество выполнения проектной работы	проект
Linux / Git для начинающих	Количество правильных ответов	тестирование
Введение в алгоритмы и структуры данных	Количество правильных ответов	тестирование
Docker и CI/CD для начинающих	Количество правильных ответов	тестирование
Итоговая проектная работа	Качество выполнения проектной работы	проект

### 7. Составители программы:

**Райгородский Андрей Михайлович**

Доктор физико-математических наук, директор ФПМИ МФТИ

**Благодарный Евгений Владимирович**

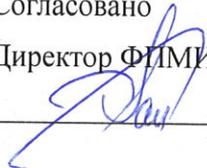
заведующий учебно-методической лабораторией инноватики

**Иванова Анастасия Сергеевна**

руководитель проектов учебно-методической лаборатории инноватики

Согласовано

Директор ФПМИ, д.ф.-м.н.

  
\_\_\_\_\_ А.М. Райгородский

**Пояснительная записка  
к разработке и реализации дополнительной программы  
профессиональной переподготовки**

**«Разработчик C++ Junior»**

Целью программы профессиональной переподготовки является получение навыков разработки программного обеспечения на языке C++, на технологиях, популярных в среде разработчиков C++. Будут рассмотрены основы программирования на C++, средства разработки, сборки и тестирования программ, основные алгоритмы и структуры данных, современные инструменты разработки программного обеспечения.

Данный курс предназначен для слушателей с отсутствующим опытом программирования и создания приложений на языке программирования C++. Участники получают основные сведения об объектно-ориентированном программировании, терминологии и синтаксисе языка, научатся создавать простые программы на языке программирования C++, изучат алгоритмы и структуры данных, инструменты Linux, Git, Docker, технологии и инструменты CI/CD.

Важную часть программы данного курса составляют практические задания, поэтому учащиеся смогут получить реальный опыт разработки программного обеспечения.

В результате обучения слушатели освоят:

- базовые понятия в программировании;
- принципы написания эффективного и безопасного кода на C++;
- основные концепции ООП;
- алгоритмы и структуры данных;
- работу с операционной системой Linux;
- работу с системой контроля версий GIT;
- инструмент контейнеризации Docker;
- инструменты CI/CD.

Задачами профессиональной переподготовки являются:

- приобретение студентами навыков проектирования и реализации приложений на языке C++ с использованием приемов объектно-ориентированного программирования;
- овладение студентами современных практик разработки: использование IDE, системы контроля версий, инструментов контейнеризации, инструментов CI/CD.

Целевая аудитория программы:

- специалисты без опыта программирования;
- специалисты, желающие переквалифицироваться в разработчиков программного обеспечения на языке C++.

Программа будет построена по очному принципу с применением дистанционных технологий (с применением электронного обучения и дистанционных образовательных технологий) в формате вебинаров, которые займут 272 ак.ч.

Преподавателями программы будут являться преподаватели МФТИ, имеющие опыт преподавания и

консультирования по заявленным модулям, а также опыт практической работы в отечественных и/или зарубежных организациях в сфере деятельности, совпадающей с направлением преподавания.

**Составители программы:**

**Райгородский Андрей Михайлович**

доктор физико-математических наук, директор ФПМИ МФТИ

**Благодарный Евгений Владимирович**

заведующий учебно-методической лабораторией инноватики

**Ивченко Олег Николаевич**

старший преподаватель кафедры АТП ФПМИ

**Иванова Анастасия Сергеевна**

руководитель проектов учебно-методической лаборатории инноватики ФПМИ

Директор ФПМИ, д.ф.-м.н.



А.М. Райгородский

**КВАЛИФИКАЦИЯ И ОПЫТ ПРИВЛЕКАЕМОГО ПРЕПОДАВАТЕЛЬНОГО СОСТАВА ДЛЯ РЕАЛИЗАЦИИ ПРОГРАММЫ  
ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ «Разработчик C++ Junior»**

<p>Ф.И.О. лектора, год рождения</p>	<p>Информация об образовании, полученном в соответствии с образовательными программами высшего профессионального образования, дополнительного профессионального образования (в т.ч. о наличии званий и ученых степеней) и т.д.</p>	<p>Место работы, занимаемая должность в настоящий момент, общий трудовой стаж, педагогический стаж</p>	<p>Опыт преподавания и консультирования по предмету, согласующемуся с направлением логот (перечислить), преподавательский стаж</p>	<p>Наличие опыта практической работы в отечественных и зарубежных организациях в сфере деятельности, совпадающей с направлением преподавания</p>
<p>Ибрагимов Булат Ленарович, 1996</p>	<p>Закончил бакалавриат (2018) и магистратуру (2020) Физтех-школы прикладной математики и информатики МФТИ, кафедра анализа данных. Выпускник школы анализа данных(2019). Аспирант МФТИ.</p>	<p>Институт Искусственного Интеллекта (AIRI), научный сотрудник. Трудовой стаж 5 лет, опыт преподавания 7 лет.</p>	<p>Преподаватель курсов по C++ и алгоритмам и структурам данных в МФТИ. Педагогический стаж 7 лет.</p>	<p>Работал разработчиком-исследователем (C++, Python) в Яндекс и Сбербанк.</p>
<p>Чернецкий Аркадий Михайлович, 1992</p>	<p>Факультет инноваций и высоких технологий (ФИБТ) МФТИ, Прикладная математика и информатика (бакалавриат, 2016)</p>	<p>Преподаватель кафедры алгоритмов и технологий программирования ФПМИ МФТИ. Общий трудовой и педагогический стаж 6 лет.</p>	<p>Преподаватель курсов «Разработка на C++», «Алгоритмы и структуры данных», «Технологии программирования», «Параллельные и распределенные вычисления» кафедры алгоритмов и технологий программирования ФПМИ МФТИ. Общий педагогический стаж 6 лет.</p>	<p>Администратор инфраструктуры кафедры алгоритмов и технологий программирования ФПМИ МФТИ.</p>
<p>Кисляков Иван Дмитриевич, 2001</p>	<p>Физтех-школы прикладной математики и информатики МФТИ, ПМИ, кафедра дискретной математики (бакалавриат).</p>	<p>ГБОУ Школа 67, ПДО. Общий трудовой и педагогический стаж 1,5 года.</p>	<p>Преподаватель курса по алгоритмам и структурам данных в ГБОУ Школа 67 и Лицее Яндекса. Педагогический стаж 1,5 года.</p>	<p>ООО «Мир технологий», отдел разработки ПО, инженер-программист</p>

<p>Рябухин Илья Алексеевич, 1999</p>	<p>Бакалавриат (2021), специальность Software Engineering, ВУЗ Czech Technical University in Prague. Магистратура (2023), Prague University of Economics and Business, специальность Information Systems Management. Дополнительное образование: акселератор Физтех Старт (1 волна, 2017), интенсив по ТехПреду от МФТИ</p>	<p>DHL IT Services, Associate Consultant. Общий трудовой и педагогический стаж 4 года.</p>	<p>Преподаватель ФКН НИУ ВШЭ с 2021, дисциплина «Научно-исследовательский семинар «Введение в облачные технологии»». Преподавательский стаж 2 года.</p>	<p>DHL IT Services, Associate Consultant.</p>
--	---	--	---	---