

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Ливанов Дмитрий Викторович
Должность: Ректор
Дата подписания: 19.03.2026 09:52:07
Уникальный программный ключ:
с6d909c49c1d2034fa3a0156c4eaa51e1b2c3a2

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 программу профессиональной
переподготовки «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 профессиональная программа
Программа профессиональной переподготовки**

«Разработчик C# Junior»

Москва 2023

Оглавление	2
1. Рабочая группа	2
2. Профессиональные стандарты, учтённые в содержании программы	2
3. Квалификационные требования, учтённые в содержании программы	2
4. Требования федерального государственного стандарта высшего образования, на основании которых была разработана программа	2
5. Характеристика новой квалификации и связанных с ней видов профессиональной деятельности, трудовых функций и (или) уровней квалификации	2
6. Характеристика компетенций, подлежащих совершенствованию, и (или) перечень компетенций, формирующихся в результате освоения программы	5
7. Цель программы	5
8. Планируемые результаты обучения по программе	6
9. Учебный план	7
10. Календарный учебный график	9
11. Рабочие программы (учебная программа)	9
12. Организационно-педагогические условия реализации дополнительной профессиональной программы	19
12.1. Требования к квалификации педагогических кадров/представителей предприятий и организаций, обеспечивающих реализацию образовательного процесса	19
12.2. Требования к материально-техническим условиям	20
12.3. Требования к информационным и учебно-методическим условиям	20
12.4. Общие требования к организации образовательного процесса	23
13. Формы аттестации и оценочные материалы по программе	24

1. Рабочая группа

С целью повышения качества дополнительной профессиональной программы (ДПП) с учетом требований профессиональных стандартов (ПС) в группу разработчиков входили:

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

2. Профессиональные стандарты, учтённые в содержании программы

Настоящая дополнительная профессиональная программа (ДПП) разработана с учётом соответствующих ей профессиональных стандартов (ПС) из национального реестра профессиональных стандартов. Профессиональные стандарты (ПС), связанные с настоящей ДПП: 06.001 Программист

3. Квалификационные требования, учтённые в содержании программы

Квалификационные требования, учтённые в содержании программы, указаны в описании должностей единого квалификационного справочника должностей руководителей, специалистов и служащих. Причём учтены требования должностей, указанных в дополнительных характеристиках, отобранных для данной программы ОТФ (п. 5. настоящего документа), которые приведены в указанных в п. 2. настоящего документа профессиональных стандартах (ПС) в строке «ЕТКС или ЕКС». Квалификационные требования инкорпорированы в настоящий документ путём отсылки и являются его неотъемлемой частью.

4. Требования федерального государственного стандарта высшего образования, на основании которых была разработана программа

Настоящей дополнительной профессиональной программе соответствует федеральный государственный стандарт высшего образования (ФГОС ВО): 01.03.02 «Прикладная математика и информатика (уровень бакалавриата)».

Настоящая дополнительная профессиональная программа разработана на основании требований указанного ФГОС ВО.

Полный перечень требований приведён в тексте указанного ФГОС ВО, инкорпорирован в настоящий документ путём отсылки и является его неотъемлемой частью.

5. Характеристика новой квалификации и связанных с ней видов профессиональной деятельности, трудовых функций и (или) уровней квалификации

«Программа профессиональной переподготовки направлена на получение компетенции, необходимой для выполнения нового вида профессиональной деятельности, приобретение новой квалификации (часть 5 статьи 76 Федерального закона N 273-ФЗ). Для их определения и может использоваться ПС. При этом необходимо учесть различия терминологии, используемой в образовании и в профессиональных стандартах, о которых говорилось выше. Вид

профессиональной деятельности, квалификация, упоминаемые в цитируемой статье закона, в ПС в большинстве случаев соответствуют ОТФ, иногда - ТФ.»¹,

Новые квалификации (ОТФ/ТФ) и их уровни:

06.001 Программист

Обобщенные трудовые функции			Трудовые функции		
код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации
А	Разработка и отладка программного кода	3	Формализация и алгоритмизация поставленных задач	А/01.3	3
			Написание программного кода с использованием языков программирования, определения и манипулирования данными	А/02.3	3
			Оформление программного кода в соответствии с установленными требованиями	А/03.3	3
			Работа с системой контроля версий	А/04.3	3
			Проверка и отладка программного кода	А/05.3	3
В	Проверка работоспособности и рефакторинг кода программного обеспечения	4	Разработка процедур проверки работоспособности и измерения характеристик программного обеспечения	В/01.4	4

¹ Письмо Министерства образования и науки РФ от 22 апреля 2015 г. N ВК-1032/06 "О направлении методических рекомендаций"

			Разработка тестовых наборов данных	V/02.4	4
			Проверка работоспособности программного обеспечения	V/03.4	4
			Рефакторинг и оптимизация программного кода	V/04.4	4
			Исправление дефектов, зафиксированных в базе данных дефектов	V/04.5	4
C	Интеграция программных модулей и компонент и проверка работоспособности выпусков программного продукта	5	Разработка процедур интеграции программных модулей	C/01.5	5
			Осуществление интеграции программных модулей и компонент и верификации выпусков программного продукта	C/02.5	5
D	Разработка требований и проектирование программного обеспечения	6	Анализ требований к программному обеспечению	D/01.6	6
			Разработка технических спецификаций на программные компоненты и их взаимодействие	D/02.6	6
			Проектирование программного обеспечения	D/03.6	6

Характеристики данных ОТФ/ТФ приведены в профессиональном стандарте (ПС), указанном в п. 2. настоящего документа. инкорпорированы в настоящий документ путём отсылки, и являются его неотъемлемой частью.

Связанные виды профессиональной деятельности (ВПД):

научно-исследовательская;

производственно-технологическая;

проектная.

Характеристика данных ВПД приведена в федеральном государственном стандарте высшего образования (ФГОС ВО), указанном в п. 4. настоящего документа, инкорпорирована в настоящий документ путём отсылки, и является его неотъемлемой частью. Указанные ВПД, освоение которых предусмотрено указанным ФГОС ВО, являются «сквозными» и служат основой овладения выбранной квалификацией (ОТФ/ТФ).

ТФ, связанные с указанными в настоящем пункте ОТФ (в случае их указания), а также их характеристики приведены в профессиональном стандарте (ПС), указанном в п. 2. настоящего документа. инкорпорированы в настоящий документ путём отсылки, и являются его неотъемлемой частью.

6. Характеристика компетенций, подлежащих совершенствованию, и (или) перечень компетенций, формирующихся в результате освоения программы

Учитывая, что слушатели программы уже освоили или осваивают основную профессиональную образовательную программу, в том числе общие компетенции, соответствующего уровня профессионального образования, компетенции формирующиеся в результате освоения настоящей программы и компетенции, подлежащие совершенствованию, а также их характеристика, представлены в ФГОС, указанном в п. 4. настоящего документа, инкорпорированы в настоящий документ путём отсылки, и являются его неотъемлемой частью.

Данные компетенции указанного ФГОС ВО служат основой определения результатов обучения по программе.

7. Цель программы

Реализация программы профессиональной переподготовки направлена на получение компетенции, необходимой для выполнения нового вида профессиональной деятельности, приобретение новой квалификации.

В соответствии с Письмом Министерства образования и науки РФ от 22 апреля 2015 г. N ВК-1032/06 «О направлении методических рекомендаций», цель представляет собой осознанное представление (предвосхищение) результата деятельности. Планируемые результаты отражены ниже.

8. Планируемые результаты обучения по программе

Планируемые результаты обучения соответствуют результатам освоения соответствующей основной профессиональной образовательной программы, а также направлены на приобретение новой квалификации, требующей изменение направленности (профиля) или специализации в рамках направления подготовки (специальности) полученного ранее профессионального образования, определены на основе профессиональных компетенций соответствующего федерального государственного образовательного стандарта, указанного в п. 4. настоящего документа.

Имеющаяся квалификация и (или) уровень образования (требования к слушателям): Высшее, незаконченное высшее образование или среднее профессиональное

Новые виды профессиональной деятельности, и ПК, соответствующие им, которые планируются к получению в результате обучения по настоящей программе, а также их характеристики, представлены в ФГОС, указанном в п. 4. настоящего документа, инкорпорированы в настоящий документ путём отсылки, и являются его неотъемлемой частью.

Выпускник также должен обладать прочими компетенциями в соответствии с указанным ФГОС, которые приведены и охарактеризованы в нём, инкорпорированы в настоящий документ путём отсылки, и являются его неотъемлемой частью.

Новые квалификации (ОТФ/ТФ), планируемые к приобретению в результате обучения по настоящей программе, приведены и охарактеризованы в п. 5. настоящего документа.

Приведённая информация о цели и результатах обучения является основой для разработки рабочих программ, оценочных материалов и иных компонентов дополнительной профессиональной программы.

9. Учебный план

№ п/п*	Наименование Модулей, дисциплин	Всего, час.	В том числе:				Форма контроля
			Лекции	Практические занятия (семинары)	Самостоятельная работа	Контрольные задания	
1	Модуль 1. Основы программирования	60	12	12	30	6	проект
2	Модуль 2. Базы данных и SQL	100	10	5	70	15	проект
3	Модуль 3. Linux / Git для начинающих	20	7	6	5	2	тестирование
4	Модуль 4. Разработка на C#	200	30	30	130	10	проект
5	Модуль 5. Введение в алгоритмы и структуры данных	80	16	16	40	8	тестирование
6	Модуль 6. Docker и CI/CD для начинающих	20	2	1	10	7	тестирование
7	Итоговая квалификационная работа	65			65		проект
	ИТОГО:	545	77	70	350	48	

* Слушатели могут проходить модули в произвольном порядке.

Срок освоения настоящей программы профессиональной переподготовки превышает минимально допустимый срок освоения 250 часов (либо другой установленный актуальными нормативными документами соответствующего Министерства срок).

10. Календарный учебный график

Модуль 1. Основы программирования

Обучение по установленной форме: *очная, с применением электронного обучения и дистанционных образовательных технологий*

Количество академических часов: 60

Порядковый № занятия	1	2	3	4	5	6
Лекции	2	2	2	2	2	2
Практические занятия	2	2	2	2	2	2
Самостоятельная работа	5	5	5	5	5	5
Контрольные задания	1	1	1	1	1	1
Итого	10	10	10	10	10	10

Модуль 2. Базы данных и SQL

Обучение по установленной форме: *очная, с применением электронного обучения и дистанционных образовательных технологий*

Количество академических часов: 100

Порядковый № занятия (по горизонтали)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
Лекции	0,5	0,5	1	0,5	0,5	0,5	0,5	1	0,5	0,5	0,5	0,5	0,5	1	0,5	0,5	0,5	0,5
Практические занятия	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	2
Самостоятельная работа	4	4	4	4	4	4	5	4	4	4	4	4	4	4	4	4	5	4
Контрольные задания	0,5	1	1	1	1	1	1	0,5	1	1	1	1	0,5	1	1	1	1	0,5
Итого	6	5,5	6	5,5	5,5	5,5	6,5	7,5	5,5	5,5	5,5	5,5	5,5	5,5	5,5	6,5	7	

Модуль 3. Linux / Git для начинающих

Обучение по установленной форме: *заочная, с применением электронного обучения и дистанционных образовательных технологий*

Количество академических часов: 20

Порядковый № занятия	1	2	3	4
Лекции	1	2,5	1,5	2
практические	1	2,5	1	1,5
самостоятельные	1	1	1	2
Контрольные задания	0,5	0,5	0,5	0,5
Итого	3,5	6,5	4	6

Модуль 4. Разработка на С#

Обучение по установленной форме: очная, с применением дистанционных технологий																
Количество академических часов: 200																
Порядковый № занятия	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Лекции	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0
практические	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0
самостоятельные	6	7	6	7	6	6	5	6	5	6	7	6	7	6	7	0
контрольные задания	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	1	0,5	0,5	0,5	0,5	1	2	37
Итого	10,5	11,5	10,5	11,5	10,5	10,5	9,5	10,5	10	10,5	11,5	10,5	11,5	11	13	37

Модуль 5. Введение в алгоритмы и структуры данных

Обучение по установленной форме: очная, с применением электронного обучения и дистанционных образовательных технологий								
Количество академических часов: 80								
Порядковый № занятия	1	2	3	4	5	6	7	8
Лекции	2	2	2	2	2	2	2	2
Практические занятия	2	2	2	2	2	2	2	2
Самостоятельная работа	5	5	5	5	5	5	5	5
Контрольные задания	1	1	1	1	1	1	1	1
Итого	10	10	10	10	10	10	10	10

Модуль 6. Docker и CI/CD для начинающих

Обучение по установленной форме: заочная, с применением электронного обучения и дистанционных образовательных технологий			
Количество академических часов: 20			
Порядковый № занятия	1	2	3
Лекции	1	0,5	0,5
практические	0,5	0	0,5
самостоятельные	3	3	4
Контрольные задания	2	2	3
Итого	6,5	5,5	8

Итоговая квалификационная работа

	Обучение по установленной форме: очная, с применением электронного обучения и дистанционных образовательных технологий
	Количество академических часов: 65
самостоятельная работа	65
Итого	65

11. Рабочие программы (учебная программа)

Модуль 1. Основы программирования

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

	4. Операции break, continue. 5. Цикл for, зачем нужен цикл счетчик.	
	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
		10
3.	Введение в коллекции. Вложенные списки	
	Теория: 1. Списки. Операции со списками и со строками. Индексация списков 2. Хранение данных, ввод данных. 3. Список списков. Хранение данных в памяти компьютера. 4. Срезки строк и списков.	2
	Практика: решение задач на списки.	2
	Самостоятельная работа.	5
	Выполнение контрольных заданий.	1
		10
4.	Функции	
	Теория: 1. Понятие функции и функционального программирования. Принимаемое константное число аргументов функции. 2. Область видимости переменных. Копирование или неkopирование элементов. Возвращаемые значения. 3. Вызов функции в памяти компьютера. Переменное число аргументов. 4. Лямбда функции.	2
	Практика: решение задач на функции.	2
	Самостоятельная работа.	5
	Выполнение контрольных заданий.	1
		10
5.	Введение в ООП	
	Теория:	2

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

Модуль 2. Базы данных и SQL

№ п/п	Содержание обучения, наименование и тематика практических занятий (вебинаров), самостоятельных работы	Объем, ак.час.
1.	Модуль I. Работа с базами данных с помощью SQL	80
1.1.	Основные понятия и базовый синтаксис выборки	6
1.2.	Создание таблицы и заполнение данными, ограничения на поля	5,5

1.3.	Строковые функции	6
1.4.	Изменение и удаление данных и UNION	5,5
1.5.	inner join и подзапросы	5,5
1.6.	другие типы join	5,5
1.7.	Агрегация	6,5
1.8.	Группировка	7,5
1.9.	Проектирование базы данных	5,5
1.10.	Работа со временем	5,5
1.11.	Оконные функции	5,5
1.12.	Логическое удаление	5,5
1.13.	Хранение истории	5,5
2.	Модуль II. NoSQL базы данных	20
2.1.	Знакомство с MongoDB	5,5
2.2.	Изменение данных в MongoDB	5,5
2.3.	Агрегация в MongoDB	6,5
2.4.	Соединение коллекций	7

Модуль 3. Linux / Git для начинающих

№ п/п	Содержание обучения, наименование и тематика практических занятий (вебинаров), самостоятельных работы	Объем, ак.час.
1	Базовый WSL и VirtualBox	3,5
	Теория: 1. Обзор курса. План занятий	1

	2. Введение. Чем отличается Windows и Linux 3. Установка WSL. Запуск 4. VirtualBox установка и использование. Тест	
	Практическая работа по теме лекции	1
	Самостоятельная работа	1
	Выполнение контрольных заданий	0,5
		6,5
2	Основы Linux	
	Теория: <ol style="list-style-type: none"> 1. Введение. Какие задачи он решает. Дистрибутив Ubuntu 2. Консоль. Основа работы с Linux. tmux 3. Скрипты. Продвинутая работа с консолью 4. Файловая система 5. Работа с сетью 6. Серверное взаимодействие. SSH 	2,5
	Практическая работа по теме лекции	2,5
	Самостоятельная работа	1
	Выполнение контрольных заданий	0,5
		4
3	WSL подробно	
	Теория: <ol style="list-style-type: none"> 1. Краткий экскурс в историю появления. WSL1 и WSL 2 2. Основные команды работы с WSL 3. Файловая система 4. Сеть WSL 5. WSL USB 6. CUDA и Windows Докер 	1,5
	Практическая работа по теме лекции	1
	Самостоятельная работа	1
	Выполнение контрольных заданий	0,5
		6
4	Git	
	Теория: <ol style="list-style-type: none"> 1. Введение. Цели. Разновидности систем контроля версий 2. Базовые основы GIT. Популярные сценарии работы 3. Серверные репозитории Git. Ветки. Merge request 4. Сценарии отката изменений. Stash 	2
	Практическая работа по теме лекции	1,5

	Самостоятельная работа	2
	Выполнение контрольных заданий	0,5
	Итого	20

Модуль 4. Разработка на C#

№ п/п	Содержание обучения, наименование и тематика практических занятий (вебинаров), самостоятельных работ	Объем, ак. час.
1	Модуль 1. Основы программирования на языке C#	44
1.1.	Занятие 1. Введение в C#.	10,5
1.2.	Занятие 2. Условные выражения и конструкции. Циклы.	11,5
1.3.	Занятие 3. Массивы. Методы.	10,5
1.4	Занятие 4. Введение в ООП. Классы и объекты.	11,5
2.	Модуль 2. Объектно-ориентированное конструирование программных систем	51
2.1	Занятие 5. Основные принципы ООП.	10,5
2.2	Занятие 6. Отношения между классами и объектами. Интерфейсы.	10,5
2.3	Занятие 7. Универсальные шаблоны в .NET.	9,5
2.4	Занятие 8. Делегаты, события и лямбда-выражения.	10,5
2.5	Занятие 9. SOLID: пять принципов объектно-ориентированного программирования.	10
3.	Модуль 3. Архитектурные паттерны. Принципы построения многослойных приложений	68

3.1	Занятие 10. Понятие архитектуры ИС. Основные термины и понятия. Основные цели и задачи курса.	10,5
3.2.	Занятие 11. Построение слоя доступа к данным (DAL). Паттерн Repository.	11,5
3.3.	Занятие 12. Инверсия управления.	10,5
3.4.	Занятие 13. MV-паттерны: MVC, MVP.	11,5
3.5.	Занятие 14. MV-паттерны: MVVM.	11
3.6.	Занятие 15. Шаблоны проектирования.	13
4	Итоговая аттестация	37
	Итого	200

Модуль 5. Введение в алгоритмы и структуры данных

№ п/п	Содержание обучения, наименование и тематика практических занятий (вебинаров), самостоятельных работы	Объем, ак.час.
1.	Введение	10
	Теория: 1. Орг. вопросы, Яндекс-контест, работа в системе, O-семантика, асимптотические классы, Мастер-теорема. 2. Обзор контейнеров в разных языках программирования, основные операции и их асимптотика. 3. Массивы, списки, двусвязные списки, стек и очередь.	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
		10
2.	Сортировки	

	Теория: 1. Понятие, примеры использования и свойства сортировок. 2. Квадратичные сортировки – сортировки вставками и пузырьком. 3. Merge Sort.	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
3.	Динамическое программирование. Бинарный (экспоненциальный) поиск.	10
	Теория: 1. Задача поиска числа в контейнере. Вычисление асимптотики. Бинарный поиск - зачем использовать. 2. Бинарный и экспоненциальный поиск. 3. Динамическое программирование - что это и зачем надо. Задача о рюкзаке. (Не)точность дп. 4. Задача о рюкзаке с дополнительным условием. Двумерное динамическое программирование.	2
	Практика: Бинарный и экспоненциальный поиск - реализация.	2
	Самостоятельная работа: Решение задач на бинарный поиск. Динамическое программирование одномерное и двумерное.	5
	Выполнение контрольных заданий.	1
4.	Хеширование. Метод цепочек.	10
	Теория: 1. Хеш-функция. Ее свойства, реализация для различных типов данных. Проблема коллизии. 2. Хеш-таблица. Контейнеры в различных языках, использующие хеш-таблицу. Алгоритм работы хеш-таблицы. 3. ООП, атрибуты класса хеш-таблицы. Реализация хеш-таблицы методом закрытой адресации. Расчет ее асимптотик.	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
5.	Хеш-таблицы с открытой адресацией	10
	Теория: 1. Разрешение коллизии методом двойного хеширования. 2. Две хеш-функции, их свойства. Реализация алгоритма.	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
6.	Куча	10
	Теория: 1. Определение и свойства кучи. 2. Вставка, Поиск максимума, Удаление. Реализация абстрактного типа данных. 3. Эффективное построение кучи. Взятие максимума/минимума. Какие задачи решает - в чем эффективность кучи. 4. Пирамидальная сортировка.	2
	Практика: Построение и реализация кучи.	2
	Самостоятельная работа: Построение кучи, взятие максимума/минимума, пирамидальная сортировка.	5
	Выполнение контрольных заданий.	1
7.	Простые деревья и граф	10
	Теория: 1. Определение дерева. Высота дерева, родительские, дочерние узлы. Способы хранения дерева. 2. Обход в глубину и ширину. 3. Поиск ключа. Вставка/удаление. Проблема несбалансированности дерева. Оценка Эффективности. 4. Решаемые задачи. Поиск компонент связности, проверка на двудольность. Поиск цикла.	2
	Практика: Реализация обхода. Решение задач по теме.	2

	Самостоятельная работа: DFS, BFS, решение задач с урока.	5
	Выполнение контрольных заданий.	1
8.	Сбалансированные деревья	10
	Теория: 1. Напоминание: зачем балансировать деревья. Мотивация использовать графы. Определение AVL-дерева. 2. AVL - дерево. 3. Вращения дерева. Операции вставки и удаления. 4. Оценка эффективности AVL-дерева.	2
	Практика: Реализация конструкции класса. Реализация алгоритма AVL-дерева и ответы на вопросы.	2
	Самостоятельная работа: Реализация AVL-дерева.	5
	Выполнение контрольных заданий.	1
	Итого	80

Модуль 6. Docker и CI/CD для начинающих

№ п/п	Содержание обучения, наименование и тематика практических занятий (вебинаров), самостоятельных работы	Объем, ак.час.
1.	Docker	6,5
	Теория: 1. Введение 2. Что такое докер? Как его используют в организациях? Лицензирование и аналоги 3. Принципы работы докер. Докер vs VM 4. Контейнер vs образ. Dockerfile 5. Работа с контейнером. docker exec 6. Создание внешних дисков, подключение директорий 7. Создание сетей. Типы сетей 8. Docker Best Practices	1
	Практическая работа по теме лекции	0,5
	Самостоятельная работа	3
	Выполнение контрольных заданий	2

2.	Docker Compose	5,5
	Теория: 1. Что такое Docker Compose? Примеры и преимущества 2. Docker Compose. Сеть и диски в docker compose	0,5
	Практическая работа по теме лекции	0
	Самостоятельная работа	3
	Выполнение контрольных заданий	2
3.	DevOps. GitLab CI	8
	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
	Итого	20

7.	Итоговая квалификационная работа	65
	Самостоятельная работа.	65
	Итого	545

Содержание дополнительной профессиональной программы направлено на достижение результатов её целей (планируемых результатов).

12. Организационно-педагогические условия реализации дополнительной профессиональной программы

12.1. Требования к квалификации педагогических кадров/представителей предприятий и организаций, обеспечивающих реализацию образовательного процесса

Требования к образованию педагогических и иных работников, а также (при наличии) требования к освоению ими дополнительных профессиональных программ, опыту работы в

области профессиональной деятельности, соответствующей направленности ДПП: высшее образование, трудовой и (или) педагогический стаж от полугода, наличие опыта практической работы в сфере деятельности, совпадающей с направлением преподавания.

12.2. 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	<p>Слушателю необходимо наличие доступа в сеть интернет, компьютер.</p> <p>Преподавателю курса необходимо наличие доступа администратора курса на LMS-платформе к материалам курса.</p>
Информационно-коммуникационная платформа дистанционных семинаров	Практические занятия (дистанционные семинары)	<p>Слушателю необходимо наличие доступа в сеть интернет, компьютер.</p> <p>Преподавателю курса необходимо оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.</p>
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.

12.3. 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. Волк, В. К. Базы данных. Проектирование, программирование, управление и администрирование : учебник для вузов / В. К. Волк. — 3-е изд., стер. — Санкт-Петербург : Лань, 2022. — 244 с. — ISBN 978-5-8114-9368-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/193373> (дата обращения: 29.03.2023). — Режим доступа: для авториз. пользователей.
2. Нестеров, С. А. Интеллектуальный анализ данных с использованием SQL Server / С. А. Нестеров. — Санкт-Петербург : Лань, 2023. — 160 с. — ISBN 978-5-507-45535-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/311861> (дата обращения: 29.03.2023). — Режим доступа: для авториз. пользователей.

Дополнительная литература

1. Bob Ward. SQL Server 2019 Revealed: Including Big Data Clusters and Machine Learning.
2. Грофф, Джеймс . SQL : Энциклопедия : пер. с англ. / Д. Р. Грофф, П. Н. Вайнберг. — 3-е изд. — СПб. : Питер. 2016
3. Дэви Силен, Арно Мейсман, Мохамед Али. Основа Data Science и BigData. Python и наука о данных. Питер. 2017

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. Залогова, Л. А. Основы объектно-ориентированного программирования на базе языка C# : учебное пособие для вузов / Л. А. Залогова. — 3-е изд., стер. — Санкт-Петербург : Лань, 2021. — 192 с. — ISBN 978-5-8114-8481-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/176894> (дата обращения: 29.03.2023). — Режим доступа: для авториз. пользователей.

Дополнительная литература

1. Виденин С. А. Архитектура информационных систем. Учебно-методическое пособие. Красноярск: Сиб. федер. ун-т, 2018. — 72 с.
2. Виденин С. А. Шаблоны проектирования информационных систем. Учебно-методическое пособие. Красноярск: Сиб. федер. ун-т, 2018. — 148 с.
3. Head First. Изучаем C#. 4-е изд, Издательский Дом ПИТЕР, 2022. - 768 с.
4. C# 10 и .NET 6. Современная кросс-платформенная разработка. Издательский Дом ПИТЕР, 2022. - 848 с.
5. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft.NET. Framework 4.5 на языке C#. Издательский Дом ПИТЕР, 2022. - 896 с.
6. Алексей Васильев. C#. Объектно-ориентированное программирование. Учебный курс, Издательство: Питер, 2012 г.
7. Герберт Шилдт. C#: учебный курс, Издательство: Питер, 2002 г.

МОДУЛЬ 5:

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 6:

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.

12.4. Общие требования к организации образовательного процесса

Форма обучения – очная / заочная, с применением дистанционных технологий.

Режим обучения – 2 раз(а) в неделю по 2-4 академических часа.

Преподаватель проводит практические занятия дистанционно в форме вебинаров с использованием платформы ZOOM (или аналогичной).

Видеоматериалы и дополнительные материалы выкладываются на обучающую платформу Skillspase.

Самостоятельная работа выполняется слушателем в удобном для слушателя режиме.

В Таблице ниже описаны образовательные технологии.

№ п/п	Вид занятия	Форма проведения занятий	Цель
1	Лекция	Изложение материала посредством лекций, обсуждение общих вопросов по тематике курса / Самостоятельный просмотр видеолекций на обучающей платформе.	Ознакомление слушателей с базовым материалом по тематике курса
2	Практическое занятие	Выполнение практических заданий, получение обратной связи от преподавателя.	Практическое освоение теоретических знаний, Разбор ситуаций, а также углубление знаний по курсу
3	Самостоятельная работа	Самостоятельное изучение дополнительных материалов и литературы. Выполнение	Углубление знаний по курсу. Применение знаний к своей организации / проекту

		тренировочных тестов и заданий.	
4	Выполнение контрольных заданий	Выполнение тестов, проверочных заданий.	Практическое освоение теоретических знаний, контроль освоения материалов
5	Итоговая аттестация	Подготовка итоговой аттестационной работы.	Практическое освоение теоретических знаний, контроль освоения материалов. Получение практического результата в своих организациях / проектах

13. Формы аттестации и оценочные материалы по программе

Оценка качества освоения программы проводится по двухбалльной системе: «зачтено», «не зачтено» по результатам промежуточного контроля (тестирование, проверочные задания на взаимную оценку), контроля посещаемости практических занятий (вебинаров) и результатам итоговой аттестации.

Слушатель считается аттестованным в случае положительных результатов работы (не менее 70% баллов от итоговой оценки) в процессе обучения и успешной сдачи экзамена. При этом баллы за экзамены начисляются только при достижении 50% порога при прохождении каждого экзаменационного испытания. После аттестации слушатель получает оценку «отлично», если набрано не меньше 80% баллов от возможного максимума, «хорошо», если набрано не меньше 65%, «удовлетворительно», если набрано не меньше 40%, в противном случае слушатель курс не сдает.

Результат тестирования, решения проверочных заданий и написания кода проверяется автоматически системой на образовательной платформе. Итоговая аттестационная работа проверяется преподавателем

Составляющие процесса обучения, которые оцениваются в ходе обучения, и их вклад в итоговую оценку представлены в Таблице ниже.

Таблица – Составляющие процесса обучения

	Основные показатели оценки	Вклад в итоговую оценку
1	Основной курс обучения на образовательной платформе	80%
2	Итоговая аттестация	20%

Оценочные материалы:

1. Пример тестовых заданий к модулю 1 «Основы программирования»

- 1) Даны три целых числа a , b , c . Определите сумму положительных из них и выведите ее на экран.
- 2) Составить программу нахождения и вывода суммы двух наибольших из трех различных целых чисел, которые введены с экрана.
- 3) Вводятся натуральные числа до тех пор, пока не будет введено -1 или пока не будет введено число, кратное 13. Найдите среднее арифметическое всех таких чисел,

которые оканчиваются на 5. (Среднее арифметическое можно найти, разделив сумму чисел на их количество)

2. Пример заданий текущего контроля к модулю 2 «Базы данных и SQL»

1) Напишите запрос, который создаст таблицу employee с полями:

- id - целое число
- name - строка с длиной 128
- lastname - строка с длиной 128
- salary - целое число

2) В базе существует таблица с заказами (orders). В таблице следующий набор полей:

- id - идентификатор заказа
- customer_id - идентификатор покупателя
- order_total - общая сумма заказа
- product_id - идентификатор продукта
- order_date - дата заказа
- amount - количество единиц заказа

Сформировать выборку, которая для каждого заказа рассчитывает разницу между максимальной стоимостью заказа у клиента и текущей суммой заказа. В выборке должны быть поля id, customer_id, delta (рассчитанная разница).

3) В базе существует таблица users с логическим удалением. Напишите запрос, который позволит восстановить пользователя с id = 3 после логического удаления.

3. Пример заданий текущего контроля к модулю 3 «Linux / Git для начинающих»

1) Зачем терминал в Linux эмулирует(притворяется) физический терминал?

Варианты ответа:

- Для правильной работы сетевых соединений
- Для возможности параллельно с ним подключать и физические терминалы
- Для обратной совместимости со старыми компонентами ОС

2) В чём разница между git reset --mixed <file> и git commit -- <file>?

Варианты ответа:

- Никакой
- Первый возвращает состояние файла в staged к предыдущему коммиту. Второй состояния файла в рабочей папке к предыдущему коммиту
- Первый возвращает состояние файла и в staged и в рабочей папке к предыдущему коммиту. Второй только состояние файла в рабочей папке

4. Пример тестовых заданий к модулю 4 «Разработка на C#»

1) Рассчитать и вывести на экран площадь и периметр прямоугольника, стороны которого запросить у пользователя (ввод с клавиатуры).

Задача на ввод/вывод информации, арифметические операции

2) Найдите сумму «1+2+3+...+n», где n вводится пользователем с клавиатуры.

Задача на цикл «For...»

- 3) Даны числа от 35 до 87. Вывести на консоль те из них, которые при делении на 7 дают остаток 1, 2 или 5.

Задача на цикл «for (int i = начальное значение; i < конечное значение; i++)», а также на условный оператор, так как нам нужно выводить не все числа из указанного диапазона, а только те, которые отвечают определенному условию (if (a ... || ...)) { Console.WriteLine(i); }, а именно делятся на 7 и в остатке дают определенное число. Проверить остаток числа от деления можно командой «%» (пример: int a = 50 % 7; // a = 1).

- 4) Представьте, что на складе имеется определенное количество ящиков с яблоками. Когда подъезжает машина для погрузки, попросить пользователя ввести с клавиатуры число – количество ящиков, которые готова забрать машина. Цикл должен работать до тех пор, пока все ящики не отгрузят со склада. Предусмотреть тот случай, когда пользователь введёт количество ящиков больше, чем есть на складе (сообщить о том, что столько нет и отгрузить сколько есть).

Задача на цикл «while...». Организовать цикл пока ящики есть (пока переменная отвечающая за ящики больше 0), в цикле вычитать из этой переменной то количество, которое запросил пользователь с клавиатуры

5. Пример тестовых заданий к модулю 5 «Введение в алгоритмы и структуры данных»

- 1) Реализовать АТД очередь на двух АТД стеках. Реализовать методы push(), pop, size, is_empty.
- 2) Реализуйте сортировку слиянием merge_sort().
- 3) Решите задачу: Используя АТД стек, по обратной Польской нотации выражения найдите его значение.

6. Пример тестовых заданий к модулю 6 «Docker и CI/CD для начинающих»

- 1) Напишите Dockerfile к приложению на фласке. Потом разверните контейнер на порту 8000

1. Скачать python и pip
2. Запустить команду pip install flask, а потом запустить команду python main.py

Пример main.py можно найти тут:

<https://flask.palletsprojects.com/en/1.1.x/quickstart/>

- 2) Сделайте CI/CD пайплан, в котором будет
 - Stage build
 - Job docker build & push - сборка докер образа и отправка в container registry
 - Stage test
 - Job run test 1 - эмуляция запуска тестов (внутри должен быть любой непустой скрипт)
 - Job run test 2 - эмуляция запуска тестов (внутри должен быть любой непустой скрипт)
 - Job run test 3 - эмуляция запуска тестов (внутри должен быть любой непустой скрипт)

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#. Также участники программы освоят работу с базами данных при помощи языка запросов SQL, 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 объектно-ориентированного программирования: абстракция, инкапсуляция, наследование, полиморфизм;
- работу с базами данных с помощью SQL;
- алгоритмы и структуры данных;
- работу с базами данных с помощью средств ORM;
- работу с операционной системой Linux;
- работу с системой контроля версий GIT;
- проектирование приложений на C# согласно методологии MV-паттернов;
- 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 147 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

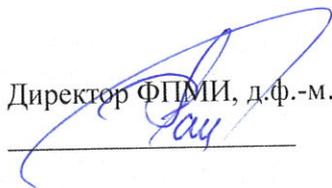
Ивченко Олег Николаевич

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 ПРЕПОДАВАТЕЛЬСКОГО СОСТАВА ДЛЯ РЕАЛИЗАЦИИ ПРОГРАММЫ
ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ «Разработчик C# Junior»**

Ф.И.О. лектора, год рождения	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.	Место работы, занимаемая должность в настоящий момент, общий трудовой стаж, педагогический стаж	Опыт преподавания и консультирования по предмету, согласующемуся с направлением лота (перечислить), преподавательский стаж	Наличие опыта практической работы в отечественных и зарубежных организациях в сфере деятельности, совпадающей с направлением преподавания
Виденин Сергей Александрович, 1983	2005 год. Красноярский государственный педагогический университет им. В.П. Астафьева. Информатики и ВТ, Учитель информатики, школьный психолог. 2009 год. Степень кандидата педагогических наук по специальности 13.00.02 – «Теория и методика обучения и воспитания (информатика)».	Национальный исследовательский университет «Высшая школа экономики», доцент факультета компьютерных наук, департамента программной инженерии. Общий трудовой и педагогический стаж более 18 лет.	Преподаватель курса по разработке на C# в МФТИ и НИУ ВШЭ, разработчик ряда учебно-методических изданий по языку C#. Педагогический стаж более 18 лет.	ООО «АстроСофт Сибирь», Ведущий программист C#. Компания Digital mind development, Программист C#.
Инанц Гайк Ашотович, 1992	2016 год. Профессиональная переподготовка. Менеджмент, Федеральная программа подготовки управленческих кадров для организаций народного хозяйства Российской Федерации. 2021 год. Доцент ВАК по специальности «Математическое и программное обеспечение вычислительных машин, компьютеров и компьютерных сетей».	МФТИ. Должность: Преподаватель	Преподавательский стаж: более 4 лет. Компании и проекты: HackerU, ITHub,	Более 5 лет опыта работы в сфере работы с данными.

		<p>Высшая школа информационных технологий и безопасности HaskelU. Должность: Преподаватель СПАО «Ингосстрах» (проект «Ингоскод»). Должность: преподаватель.</p>	<p>МФТИ, Сбербанк, МТС, Ингосстрах, Школа СТО яндекса.</p>	<p>Работа над КХД таких банков, как: ВТБ, Райффайзен банк, Альфа банк.</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 С++», «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 TechПреду 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>