

Министерство науки и высшего образования Российской Федерации
федерально государственное автономное образовательное учреждение
высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»
(МФТИ, Физтех)

УТВЕРЖДАЮ

Ректор МФТИ

д-р физ.-мат. наук, профессор

Д.В. Ливанов



«29» февраля 2024 г.

**Дополнительная профессиональная программа
профессиональной переподготовки
«Программирование на Python»**

УГСН 09.00.00 Информатика и вычислительная техника

Направление подготовки 09.03.01 Информатика и вычислительная техника

ОКВЭД 62.01 Разработка компьютерного программного обеспечения

Москва 2024

Оглавление

РАЗДЕЛ 1. ХАРАКТЕРИСТИКА ПРОГРАММЫ	6
1. Общая характеристика программы	6
1.1 Цель реализации программы	6
1.2. Характеристика новой квалификации и связанных с ней видом профессиональной деятельности, трудовых функций и/или уровней квалификации	7
1.3 Совершенствуемые и/или приобретаемые компетенции	8
1.4. Планируемые результаты обучения	10
1.5 Требования к уровню подготовки поступающего на обучение, необходимому для освоения программы	21
1.6 Форма, объем и режим обучения	21
РАЗДЕЛ 2. СОДЕРЖАНИЕ ПРОГРАММЫ	22
2.1. Учебный (тематический) план	22
2.2. Календарный учебный график	25
2.3. Рабочие программы дисциплин (модулей)	26
РАБОЧАЯ ПРОГРАММА МОДУЛЯ 1	27
1. Цель освоения Модуля 1	27
1.1. Совершенствуемые и/или приобретаемые компетенции	27
1.2. Планируемые результаты обучения	28
1.3. Категория обучающихся	31
1.4. Форма обучения	31
1.5. Объем программы	31
1.6. Режим обучения	31
2. Содержание программы	32
2.1. Учебный (тематический) план	32
2.2. Учебная программа	41
3. Формы аттестации и оценочные материалы	47
3.1. Формы аттестации	47
3.2. Оценочные материалы	48
4. Организационно-педагогические условия реализации программы	53
4.1. Учебно-методическое обеспечение и информационное обеспечение программы 53	
4.1.1. Список литературы	53
4.1.2. Ресурсы информационно-телекоммуникационной сети "Интернет"	53
4.2. Описание материально-технической базы, необходимой для осуществления образовательного процесса по программе	54
5. Организация образовательного процесса	55
РАБОЧАЯ ПРОГРАММА МОДУЛЯ 2	56

1. Цель освоения Модуля 2.....	56
1.1. Совершенствуемые и/или приобретаемые компетенции	56
1.2. Планируемые результаты обучения	57
1.3. Категория обучающихся.....	60
1.4. Форма обучения	60
1.5. Объем программы	60
1.6. Режим обучения	60
2. Содержание программы.....	60
2.1. Учебный (тематический) план	60
2.2. Учебная программа	66
3. Формы аттестации и оценочные материалы.....	68
3.1. Формы аттестации.....	68
3.2. Оценочные материалы	69
4. Организационно-педагогические условия реализации программы	72
4.1. Учебно-методическое обеспечение и информационное обеспечение программы	72
4.1.1. Список литературы.....	72
4.1.2. Ресурсы информационно-телекоммуникационной сети "Интернет"	72
4.2. Описание материально-технической базы, необходимой для осуществления образовательного процесса по программе	73
5. Организация образовательного процесса.....	73
РАБОЧАЯ ПРОГРАММА МОДУЛЯ 3	74
1. Цель освоения Модуля 3.....	74
1.1. Совершенствуемые и/или приобретаемые компетенции	74
1.2. Планируемые результаты обучения	75
1.3. Категория обучающихся.....	77
1.4. Форма обучения	77
1.5. Объем программы	77
1.6. Режим обучения	77
2. Содержание программы.....	77
2.1. Учебный (тематический) план	77
2.2. Учебная программа	81
3. Формы аттестации и оценочные материалы.....	84
3.1. Формы аттестации.....	84
3.2. Оценочные материалы	85
4. Организационно-педагогические условия реализации программы	88
4.1. Учебно-методическое обеспечение и информационное обеспечение программы	88

4.1.1.	Список литературы.....	88
4.1.2.	Ресурсы информационно-телекоммуникационной сети "Интернет"	88
4.2.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по программе	88
5.	Организация образовательного процесса.....	89
РАБОЧАЯ ПРОГРАММА МОДУЛЯ 4.....		90
1.	Цель освоения Модуля 4.....	90
1.1.	Совершенствуемые и/или приобретаемые компетенции	90
1.2.	Планируемые результаты обучения	91
1.3.	Категория обучающихся.....	94
1.4.	Форма обучения	94
1.5.	Объем программы	94
1.6.	Режим обучения	94
2.	Содержание программы.....	94
2.1.	Учебный (тематический) план	94
2.2.	Учебная программа	98
3.	Формы аттестации и оценочные материалы.....	100
3.1.	Формы аттестации.....	100
3.2.	Оценочные материалы.....	101
4.	Организационно-педагогические условия реализации программы	104
4.1.	Учебно-методическое обеспечение и информационное обеспечение программы 104	
4.1.1.	Список литературы.....	104
4.1.2.	Ресурсы информационно-телекоммуникационной сети "Интернет"	104
4.2.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по программе	105
5.	Организация образовательного процесса.....	105
РАЗДЕЛ 3. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ		107
РАЗДЕЛ 4. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ		109
4.1.	Материально-технические условия и организационное обеспечение реализации дополнительной профессиональной программы профессиональной переподготовки ..	109
4.2.	Учебно-методическое обеспечение реализации дополнительной профессиональной программы профессиональной переподготовки	110
4.2.1.	Учебно-методическое обеспечение и информационное обеспечение программы.....	110
4.2.2.	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю).....	111
4.2.3.	Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)	111

5. Составители программы	113
--------------------------------	-----

РАЗДЕЛ 1. ХАРАКТЕРИСТИКА ПРОГРАММЫ

«Программирование на Python»

1. Общая характеристика программы

Реализация программы профессиональной переподготовки направлена на получение компетенции, необходимой для выполнения нового вида профессиональной деятельности. Программа профессиональной переподготовки разрабатывается на основании установленных квалификационных требований к должности «Инженер-программист (программист)», профессионального стандарта 06.001 «Программист» и требований федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.01 «Информатика и вычислительная техника» к результатам освоения образовательной программы.

Дополнительная профессиональная программа профессиональной переподготовки является модульной. Каждый модуль представляет собой завершённую программу, направленную на формирование/совершенствование у слушателей конкретных компетенций. Объём программы каждого модуля превышает 16 академических часов. Обучение по каждому модулю завершается промежуточной (итоговой аттестацией) для проверки сформированности заявленных компетенций. При условии успешного прохождения промежуточной (итоговой) аттестации по модулю слушателю выдается удостоверение о повышении квалификации установленного образца.

Дополнительная профессиональная программа профессиональной переподготовки включает прохождение практики в организации партнера в дистанционном формате

Слушателям, успешно прошедшим итоговую аттестацию по дополнительной профессиональной программе профессиональной переподготовки, выдается диплом с присвоением квалификации «Программист» установленного образца.

1.1 Цель реализации программы

Целью реализации дополнительной профессиональной программы профессиональной переподготовки «Программирование на Python» является формирование/совершенствование компетенций слушателей в области решения профессиональных задач по программированию с использованием языка Python, применения шаблонов проектирования на Python, работы с Python библиотеками, применения объектно-ориентированного и функционального программирования.

1.2. Характеристика новой квалификации и связанных с ней видов профессиональной деятельности, трудовых функций и/или уровней квалификации

Настоящая дополнительная профессиональная программа профессиональной переподготовки разработана с учётом соответствующих ей профессиональных стандартов и направлена на получение компетенций, необходимой для выполнения нового вида профессиональной деятельности «Программист».

Таблица 1

Профессиональный стандарт	
06.001 Программист	
Обобщенная трудовая функция	
А. Разработка и отладка программного кода	
Трудовая функция	Компетенции
1. Написание программного кода с использованием языков программирования, определения и манипулирования данными	ПК-1
2. Работа с системой контроля версий	ПК-2
3. Проверка и отладка программного кода	ПК-5

Программа профессиональной переподготовки разрабатывается на основании квалификационных требований, установленных Единым квалификационным справочником должностей руководителей, специалистов и служащих (ЕКС).

Настоящая дополнительная профессиональная программа разработана в соответствии с ФГОС ВО 09.03.01 Информатика и вычислительная техника (уровень бакалавриата).

Программа нацелена на студентов, обучающихся по следующим направлениям/специальностям:

- 03.04.01 Прикладные математика и физика
- 16.04.01 Техническая физика
- 14.04.02 Ядерная физика и технологии
- 06.04.01 Биология
- 22.04.01 Материаловедение и технологии материалов.

1.3 Совершенствуемые и/или приобретаемые компетенции

Компетенции, формируемые и совершенствуемые в результате обучения, представлены в таблицах 2, 3 и 4.

Таблица 2

№	Компетенция в соответствии с профессиональным стандартом 06.001 Программист	Код компетенции
1	Способен писать программный код с использованием языков программирования, определения и манипулирования данными	ПК-1
2	Способен работать с системой контроля версий	ПК-2
3	Способен проверить и отладить программный код	ПК-5

Таблица 3

№	Компетенции в соответствии с направлением подготовки 09.03.01 Информатика и вычислительная техника (ФГОС)	Код компетенции
1	Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8
2	Способен осваивать методики использования программных средств для решения практических задач	ОПК-9
3	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	УК-3

Таблица 4а

№	Профессиональные компетенции (ПК) в соответствии с направлением подготовки 09.03.01 Информатика и вычислительная техника (Собственный стандарт МФТИ)	Код компетенции
1.	Способен формализовать и алгоритмизировать поставленную задачу, написать программный код с использованием языков программирования, оформить код в соответствии с установленными требованиями	ПК-3

Профессиональные компетенции (ПК) в соответствии с Матрицей (моделью) цифровых компетенций в части профессиональных компетенций в ИТ-сфере (Иннополис)

Таблица 46

Наименование сферы	Код и наименование профессиональной компетенции	БАЗОВЫЙ УРОВЕНЬ РАЗВИТИЯ КОМПЕТЕНЦИЙ Способность проявляется под внешним контролем / при внешней постановке задачи/ обучающийся пользуется готовыми, рекомендованными продуктами
Средства программной разработки	ПК-4 Применяет языки программирования для решения профессиональных задач	Самостоятельно применяет языки программирования. Использует настраиваемые программные инструменты для автоматизации процессов в профессиональной деятельности.
Интернет-технологии	ПК-6 Разрабатывает различные веб-ориентированные решения	Участвует под контролем опытных специалистов в разработке веб-решений, с применением фреймворков и платформ
Средства программной разработки	ПК-7 Использует СУБД при разработке ПО	Знает основы баз данных, знаком с нормализацией, ACID, транзакциями, может написать простые выборки. Участвует в проектах по созданию ПО с использованием СУБД под контролем опытных специалистов

Перечень областей профессиональной деятельности обучающихся, в которых развиваемые цифровые компетенции будут использоваться:

- 06 Связь, информационные и коммуникационные технологии (в сфере развития фундаментальных математических и физических основ связи и информационно-коммуникационных технологий, инновационных и опытно-конструкторских разработок);
- 40 Сквозные виды профессиональной деятельности в промышленности (в сфере фундаментальных и прикладных научно-исследовательских, инновационных и опытно-конструкторских разработок, а также в сфере разработки и внедрения новых технологических процессов производства перспективных материалов (в том числе композитов, нано- и метаматериалов), изделий опто-, микро- и наноэлектроники, разработки и применения электронных приборов и комплексов, а также в сфере мониторинга параметров материалов, состояния сложных технических и живых систем и состояния окружающей среды, включая разработку и использование для решения поставленных задач)
- 32 Авиастроение (в сфере фундаментальных и прикладных исследований в области аэрофизики, аэромеханики и летательной техники, физики прочности и механики сплошных сред, материаловедения, молекулярной и химической физики, инновационных и опытно-конструкторских разработок);

- 25 Ракетно-космическая промышленность (в сфере фундаментальных и прикладных исследований, инновационных и опытно-конструкторских разработок в области ракетостроения, освоения космического пространства, аэро- и космической физики).

1.4. Планируемые результаты обучения

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 ФГОС ВО.

Планируемые результаты обучения представлены в таблице 5.

Таблица 5

	Уметь - знать	09.03.01 Информатика и вычислительная техника ФГОС ВО
		Квалификация: бакалавр
		Код компетенции
1	Знать: <ul style="list-style-type: none"> – принципы организации кода – понятия коллекций и функций в Python – назначение функций в языках программирования – сетевые термины (сокеты, клиент, сервер) – первичных ключей, типов данных атрибутов умеет строить дерево решений для выбора СУБД – принципы проектирования SOLID – смысл принципов проектирования SOLID – правильную структуру документа – номинальные, порядковые, интервальные шкалы и шкалу отношений – отличия разных типов шкал – категориальные и количественные переменные 	ОПК-8

- меры распределения переменных
- что такое частота распределения
- что такое нормальное распределение и как его интерпретировать
- свойства нормального распределения.
- правило трех сигм
- дискретное, непрерывное и равномерное распределение

Уметь:

- использовать базовые типы и конструкции Python для написания простых программ
- сопоставить и выбрать необходимую структуру данных для конкретной практической задачи
- составить иерархию классов и описать их методы для конкретной практической задачи
- выделять в задаче на естественном языке необходимость применения многопоточности
- типы структур данных, операций над данными
- составлять описание требований к будущему программному обеспечению
- пользоваться библиотекой requests
- загружать данные с расширениями csv., raw., xls.,xlsx., mat., xml., json
- определить какие переменные принадлежат к какому типу шкалы
- отличить моду, среднее, медиану
- найти / вычислить моду, среднее, медиану
- строить функции и интерпретировать полученный график
- решать задачи по построению разных видов диаграмм (точечная диаграмма, столбиковая диаграмма, диаграмма соотношения), различать нормальное распределение, определять и удалять выбросы

2	<p>Знать:</p> <ul style="list-style-type: none"> – базовые типы и конструкции Python – роль коллекций и функций в программировании на Python – принцип работы клиент-серверной архитектуры – принципы проектирования БД, применяемые в работе программ и приложений – методы модуля BS – инструменты для обработки и анализа данных – меры разброса – что такое асимметрия и эксцесс – как посчитать точечную оценку и интерпретировать ее – как рассчитать доверительный интервал и интерпретировать его – центральную предельную теорему <p>Уметь:</p> <ul style="list-style-type: none"> – выделять в задаче на естественном языке необходимость применения базовых конструкций языка – определить и спроектировать необходимые классы и методы классов для конкретной предметной области – создать программу, использующую несколько потоков, для конкретной практической задачи – запускать анимации – работать с кривой плавности – делать анимацию для нескольких объектов – с помощью библиотеки requests извлекать информацию из интернета – вычислить дисперсию, стандартное отклонение – интерпретировать результаты вычислений 	ОПК-9
---	--	-------

	<ul style="list-style-type: none"> – интерпретировать графики – вычислять проценти и квантили – решать задачи на комбинаторику, сложение и произведение вероятностей, с зависимыми событиями, случайными событиями, объяснять формулу Байеса, формулу Бернулли, решать задачи на условную вероятность, задачи на математическое ожидание, дисперсию и стандартное отклонение случайной величины, воспроизводить статистическое определение вероятности 	
3	<p>Уметь:</p> <ul style="list-style-type: none"> – выполнять проектную деятельность – построить график для демонстрации 	УК-3
		<p>Профессиональный стандарт</p> <p>06.001 Программист</p>
		Код компетенции
	<p>Знать:</p> <ul style="list-style-type: none"> – понятия классов и объектов в Python, понимает их взаимосвязь – основные паттерны ООП – в каких ситуациях применять основные паттерны ООП – что такое плейсхолдеры – правила использования плейсхолдеров <p>Уметь:</p> <ul style="list-style-type: none"> – написать простой код на Python – написать собственную функцию на языке Python – использовать магические методы в написании собственных программ на Python – писать многопоточных код на Python 	ПК-1

	<ul style="list-style-type: none"> – таблицу в БД, создавать БД через консоль, удалять таблицу из БД, выгружать БД в SQL формате, отображать все таблицы в БД, отображать структуру БД – рисовать средствами QPainter – рисовать средствами QCanvas – составлять список книг в каталоге при помощи URL – определять и заменять пропущенные значения, значения типа NULL, – создавать вектор – выполнять операции сложения и вычитания векторов – выполнять операции умножения и деления векторов – выполнять операцию векторное точечное произведение – вычислять норму векторов – вычислять расстояние между векторами – вычислять скалярное произведение и угол между векторами 	
	<p>Знать:</p> <ul style="list-style-type: none"> – понятие наследования – механизмы наследования и его роль в программировании на Python – библиотеки Python для обработки данных – термины нормализации – основные нормальные формы: 1НФ, 2НФ, 3НФ, НФБК прочие нормальные формы: 4NF, 5NF, DKNF, 6NF – основные протоколы транспортного уровня – как собирать данные из интернета – какие существуют способы для извлечения 	ПК-2

	<p>информации из интернета</p> <ul style="list-style-type: none"> – что такое URL, view, шаблоны – отличия разных расширений данных – выборку и генеральную совокупность <p>Уметь:</p> <ul style="list-style-type: none"> – создать программу по описанию задачи на естественном языке с использованием коллекций языка Python – использовать конструкции для генерации исключений на Python – добавлять записи – использовать паттерны, стандартизировать код – пользоваться модулем для подгрузки данных из интернета для решения собственных задач – выявлять данные, которым можно доверять – выгружать данные – определять правильную (достаточную) выборку – интерпретировать результаты выборки и соотносить их с результатами генеральной совокупности 	
	<p>Знать:</p> <ul style="list-style-type: none"> – где находится каталог библиотек – понятия тестирования и отладки – способы синхронизации потоков – особенности работы с глобальным шлюзом GIL – как функционирует взаимодействие в интернете – что такое WEB-API – как устроена генерация ответа на HTTP-запрос – существующие источники данных <p>Уметь:</p>	<p>ПК-5</p>

	<ul style="list-style-type: none"> – установить интерпретатор Python себе на компьютер – установить среду разработки PyCharm – читать и записывать данные из файла – устанавливать внешние библиотеки в Python – использовать Git для отслеживания истории изменений версий кода – протестировать код на Python – вручную создавать или устанавливать библиотеки – проанализировать код на Python и исправить ошибки в программе – использовать возможности библиотеки asyncio для реализации асинхронности – обращаться к записям, фильтровать поиск – вносить изменения в записи, удалять записи – устанавливать соединения и выполнение запроса. – тестировать код БД, проверять на ошибки, работать с исключениями – пользоваться WEB-API – тестировать значимость нулевой гипотезы 	
		<p style="text-align: center;">09.03.01 Информатика и вычислительная техника (Собственный стандарт МФТИ)</p> <p style="text-align: center;">Квалификация: бакалавр</p> <p style="text-align: center;">Код компетенции</p>
1	<p>Знать:</p> <ul style="list-style-type: none"> – несколько встроенных функций языка Python 	ПК-3

- понятие особых методов классов
- роль особых классов в программировании на Python
- принцип работы асинхронного взаимодействия
- как должно выглядеть представление данных в табличной форме.
- свойства таблицы стилей QSS и умеет их применять
- что такое блоки

Уметь:

- использовать коллекции и функции для написания программ на Python
- создавать классы и использовать методы классов
- создавать программу для обработки исключений на Python в условиях конкретной практической задачи
- писать код для асинхронных приложений на Python
- проектировать системы с использованием паттернов
- разрабатывать формы, ориентированные на работу с базами данных.
- создавать потоки и ориентироваться в них.
- настраивать взаимодействие с главным потоком приложения.
- управлять циклом внутри потока
- создавать окно, элементы в нем, перемещать элементы внутри окна, добавлять элементам цвет
- пользоваться блоками
- создавать массив
- вызывать элемент с помощью индекса
- добавлять элемент в массив
- добавлять новые строки и столбцы
- удалять элемент из массива
- менять элементы местами внутри массива

	<ul style="list-style-type: none"> – изменять размеры массива – транспонировать элементы массива 	
		<p>Модель цифровых компетенций (в части профессиональных компетенций в ИТ-сфере) разработанная Университетом Иннополис</p> <p>Код компетенции</p>
1	<p>Знать:</p> <ul style="list-style-type: none"> – особенности объектно-ориентированной модели в Python – механизм формирования исключений – конструкции языка Python для создания потоков – примеры асинхронного программирования – принцип работы системы Git – понятия процессов и потоков – синтаксис Python – углубленно среду для разработки, ее особенности, может использовать ее для своей работы – виджеты и сигналы – как получить данные с помощью регулярных выражений – структуру проекта на Django <p>Уметь:</p> <ul style="list-style-type: none"> – вызывать и получать результат выполнения встроенных функций Python – создать программу по описанию задачи на естественном языке с использованием функций языка Python 	ПК-4

	<ul style="list-style-type: none"> – написать программу на Python с применением библиотек для конкретной практической задачи – использовать основные функции и методы Python для написания собственных клиентских и серверных приложений – определить, требуется ли в приложении многопоточность – выполнять сложные запросы SELECT, объединять таблицы, делать подзапросы и группировать таблицы и БД – оптимизировать однотипные виджеты – создавать окна в PyQt – работать с классами Qt вне главного потока. – изменять существующие виджеты Qt и кастомизировать их – собирать данные с помощью регулярных выражений – извлекать и изменять данные при помощи регулярных выражений и модуля BeautifulSoup; – добавлять файлы проекта на Django – использовать функции <code>df.head()</code>, <code>df.info()</code>, <code>set_axis()</code>, <code>dropna()</code>, <code>isnull()</code>, <code>fillna()</code>, <code>duplicated()</code>, <code>df.dtypes()</code>, <code>to_numeric()</code>, <code>astype()</code> – создавать датафрейм и выводить его на экран. – выводить количество строк и столбцов датафрейма. – выводить количество элементов в датафрейме – проводить простейший статистический анализ датафрейма с помощью <code>describe()</code> – выбирать один элемент из датафрейма – присваивать и заменять значения элементу датафрейма – проводить операции между различными датафреймами (переносы) 	
--	--	--

	<ul style="list-style-type: none"> – пользоваться функциями read_csv(), data.pivot_table(), count(), sum(), groupby(), merge(), sort(), reset_index(), fillna() 	
2	<p>Знать:</p> <ul style="list-style-type: none"> – чем отличаются два основных веб-фреймворка на Python – какие существуют web-фреймворки на Python <p>Уметь:</p> <ul style="list-style-type: none"> – создавать веб-приложения на основе фреймворка Django; – настраивать взаимодействие с базами данных с помощью инструмента Django ORM; 	ПК-6
3	<p>Знать:</p> <ul style="list-style-type: none"> – определение и основные характеристики БД – отличия БД от таблицы в Excel – свойства БД (быстродействие, быстрота, независимость, стандартизация, безопасность, интегрированность, многопользовательский доступ) – типы БД и их отличия друг от друга (иерархические, сетевые, объектно-ориентированные, реляционные, нереляционные) – что такое реляционная база данных и СУБД Реляционная модель данных – Основные функции реляционной СУБД – Особенности проектирования реляционных БД <p>Уметь:</p> <ul style="list-style-type: none"> – перечислить функции БД – привести примеры использования БД – подключаться к БД, создавать/ удалять БД и таблицы, заполнять/ редактировать/ выводить таблицы, импортировать и экспортировать файлы, PRIMARY KEY и FOREIGN KEY – строить БД под требования приложения 	ПК-7

1.5 Требования к уровню подготовки поступающего на обучение, необходимому для освоения программы

К освоению ДПП III допускаются лица, освоившие ОПОП ВО бакалавриата – в объеме не менее первого курса (бакалавры 2-го курса), освоившие ОПОП ВО специалитета – не менее первого и второго курсов (специалисты 3-го курса), учащиеся ОПОП ВО магистратур не ИТ-профиля.

1.6 Форма, объем и режим обучения

Форма обучения: Очная с применением дистанционных образовательных технологий.

Объем программы: 270 академических часов.

Режим обучения: 9 месяцев.

РАЗДЕЛ 2. СОДЕРЖАНИЕ ПРОГРАММЫ

2.1. Учебный (тематический) план

Учебный (тематический) план программы представлен в таблице 6.

Таблица 6

№ п/п	Наименование модулей, разделов и тем	Всего, час.	Лекции	Практич. работа	Самост. работа	Форма контроля
УЧЕБНЫЙ ПЛАН						
1.	Основы программирования на Python	92	21,8	26	42,2	Зачет
2.	Объектно-ориентированное программирование, графический интерфейс и основы работы с базами данных в Python	61	15,1	18	25,9	Зачет
3.	Создание Web-сервисов на Python	62	11,6	18	30,4	Зачет
4.	Анализ данных в Python	37	5,5	14	15,5	Зачет
	Итоговая аттестация	18		10	8	Защита проекта
	ИТОГО	270	54	86	114	
УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН						
<i>Модуль 1. Основы программирования на Python</i>						
1.1	Введение в программирование на Python	15	2,8	4	8,2	Тест, задания на программирование
1.2	Структуры данных и функции	16	5,9	4	6,1	Тест, задания на программирование
1.3	Объектно-ориентированное	16	2,9	4	9,1	Тест, задания на

	программирование					программиров ание
1.4	Углубленный Python	24	6,9	6	11,1	Тест, задания на программиров ание
1.5	Работа с сетью	8	2,1	2	3,9	Тест, задания на программиров ание
1.6	Тестирование кода	7	1,2	2	3,8	Тест, задания на программиров ание
	Промежуточная (итоговая) аттестация по модулю 1	6		4	2	зачет
	Итого по модулю 1	92	21,8	26	42,2	
Модуль 2. Объектно-ориентированное программирование, графический интерфейс и основы работы с базами данных в Python						
2.1	Работа с базами данных	15	4,3	4	6,7	Тест, задания на программиров ание
2.2	Паттерны ООП на Python для разработки приложения	8	2,1	2	3,9	Тест, задания на программиров ание
2.3	Построение графических интерфейсов	32	8,7	12	15,3	Задания проекта
	Промежуточная (итоговая) аттестация по модулю 2	6		4	2	зачет
	Итого по модулю 2	61	15,1	18	25,9	
Модуль 3. Создание Web-сервисов на Python						

3.1	Общее представление о WEB	8	1,4	2	4,6	Тест
3.2	Сбор данных со сторонних сайтов	8	1,7	2	4,3	Тест, задания на программирование
3.3	Beautiful Soup и работа с API	8	2,1	2	3,9	Задание на программирование
3.4	Разработка WEB-приложения	32	6,4	8	17,6	Задания проекта
	Промежуточная (итоговая) аттестация по модулю 3	6		4	2	зачет
	Итого по модулю 3	62	11,6	18	30,4	
Модуль 4. Анализ данных в Python						
4.1	Введение в анализ данных	8	1,9	2	4,1	Тест, задания на программирование
4.2	Исследование данных с Python	16	3,1	4	8,9	Тест, задания на программирование
4.3	Проект	7	0,5	4	2,5	Проект
	Промежуточная (итоговая) аттестация по модулю 4	6		4	2	зачет
	Итого по модулю 4	37	5,5	14	15,5	
	Итоговая аттестация по программе	18		10	8	Защита проекта
	Итого:	270	54	86	114	

2.2.Календарный учебный график

Календарный учебный план составляется при сформированной группе с учетом уровня их подготовки.

Календарный учебный график отражает периоды теоретических занятий, практик, процедур промежуточной и итоговой аттестаций и т.д.

КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Таблица 7

Учебные недели	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Учебные занятия (Т)																					
Практика (П)																					
Самостоятельная работа (СР)																					
Контроль (З, Э)																					
Итоговая аттестация (А)																					

продолжение Таблицы 7

Учебные недели	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	
Учебные занятия (Т)																					
Практика (П)																					
Самостоятельная работа (СР)																					
Контроль (З, Э)																					
Итоговая аттестация (А)																					

2.3.Рабочие программы дисциплин (модулей)

Рабочие программы дисциплин (модулей) отражают дисциплинарное содержание дополнительной профессиональной программы профессиональной переподготовки.

Модуль 1. Основы программирования на Python

Модуль 2. Объектно-ориентированное программирование, графический интерфейс и основы работы с базами данных в Python

Модуль 3. Создание Web-сервисов на Python

Модуль 4. Анализ данных в Python

РАБОЧАЯ ПРОГРАММА МОДУЛЯ 1

Основы программирования на Python

1. Цель освоения Модуля 1

Целью реализации модуля «Основы программирования на Python» является совершенствование компетенций слушателей в области основ программирования на Python: решения простых задач в Python, применения шаблонов проектирования на Python, работы с Python библиотеками, базового применения объектно-ориентированного и функционального программирования.

1.1. Совершенствуемые и/или приобретаемые компетенции

Компетенции, формируемые и совершенствуемые в результате обучения, представлены в таблицах 8 и 9.

Таблица 8

№	Компетенции в соответствии с профессиональным стандартом 06.001 Программист	Код компетенции
1	Способен написать программный код с использованием языков программирования, определения и манипулирования данными	ПК-1
2	Способен работать с системой контроля версий	ПК-2
3	Способен проверить и отладить программный код	ПК-5

Таблица 9а

№	Компетенции в соответствии с направлением подготовки 09.03.01 Информатика и вычислительная техника (ФГОС)	Код компетенции
1.	Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8
2	Способен осваивать методики использования программных средств для решения практических задач	ОПК-9

Таблица 9б

№	Профессиональные компетенции (ПК) в соответствии с направлением подготовки 09.03.01 Информатика и вычислительная техника (Собственный стандарт МФТИ)	Код компетенции
1.	Способен формализовать и алгоритмизировать поставленную задачу, написать программный код с использованием языков программирования, оформить код в соответствии с установленными требованиями	ПК-3

Профессиональные компетенции (ПК) в соответствии с Матрицей (моделью) цифровых компетенций в части профессиональных компетенций в ИТ-сфере (Иннополис)

Таблица 9в

Наименование сферы	Код и наименование профессиональной компетенции	Базовый уровень развития компетенций. Компетенция проявляется в незначительной степени, по заданным шаблонам и с посторонней помощью
Средства программной разработки	ПК-4 Применяет языки программирования для решения профессиональных задач	Самостоятельно применяет языки программирования. Использует настраиваемые программные инструменты для автоматизации процессов в профессиональной деятельности.

1.2. Планируемые результаты обучения

Планируемые результаты обучения представлены в таблице 10.

Таблица 10

№	Уметь - знать	Направление подготовки 09.03.01 Информатика и вычислительная техника (ФГОС)
		Квалификация: бакалавр
		Код компетенции
1	<p>Знать:</p> <ul style="list-style-type: none"> – принципы организации кода – понятия коллекций и функций в Python – назначение функций в языках программирования – сетевые термины (сокеты, клиент, сервер) <p>Уметь:</p> <ul style="list-style-type: none"> – использовать базовые типы и конструкции Python для написания простых программ – сопоставить и выбрать необходимую структуру данных для конкретной практической задачи – составить иерархию классов и описать их методы для конкретной практической задачи – выделять в задаче на естественном языке необходимость применения многопоточности 	ОПК-8

2	<p>Знать:</p> <ul style="list-style-type: none"> – базовые типы и конструкции Python – роль коллекций и функций в программировании на Python – принцип работы клиент-серверной архитектуры <p>Уметь:</p> <ul style="list-style-type: none"> – выделять в задаче на естественном языке необходимость применения базовых конструкций языка – определить и спроектировать необходимые классы и методы классов для конкретной предметной области – создать программу, использующую несколько потоков, для конкретной практической задачи 	ОПК-9
		<p>Профессиональный стандарт 06.001 Программист</p>
3	<p>Знать:</p> <ul style="list-style-type: none"> – понятия классов и объектов в Python, понимает их взаимосвязь <p>Уметь:</p> <ul style="list-style-type: none"> – написать простой код на Python – написать собственную функцию на языке Python – использовать магические методы в написании собственных программ на Python – писать многопоточных код на Python 	ПК-1
4	<p>Знать:</p> <ul style="list-style-type: none"> – понятие наследования – механизмы наследования и его роль в программировании на Python – библиотеки Python для обработки данных <p>Уметь:</p> <ul style="list-style-type: none"> – создать программу по описанию задачи на естественном языке с использованием коллекций языка Python – использовать конструкции для генерации исключений на Python 	ПК-2

5	<p>Знать:</p> <ul style="list-style-type: none"> – где находится каталог библиотек – понятия тестирования и отладки – способы синхронизации потоков – особенности работы с глобальным шлюзом GIL <p>Уметь:</p> <ul style="list-style-type: none"> – установить интерпретатор Python себе на компьютер – установить среду разработки PyCharm – читать и записывать данные из файла – устанавливать внешние библиотеки в Python – использовать Git для отслеживания истории изменений версий кода – протестировать код на Python – вручную создавать или устанавливать библиотеки – проанализировать код на Python и исправить ошибки в программе – использовать возможности библиотеки asyncio для реализации асинхронности 	ПК-5
		<p align="center">Направление подготовки 09.03.01 Информатика и вычислительная техника (Собственный стандарт МФТИ)</p>
6	<p>Знать:</p> <ul style="list-style-type: none"> – несколько встроенных функций языка Python – понятие особых методов классов – роль особых классов в программировании на Python – принцип работы асинхронного взаимодействия <p>Уметь:</p> <ul style="list-style-type: none"> – использовать коллекции и функции для написания программ на Python – создавать классы и использовать методы классов – создавать программу для обработки исключений на Python в условиях конкретной практической задачи – писать код для асинхронных приложений на Python 	ПК-3
		<p align="center">Матрица (модель) цифровых компетенций в части профессиональных компетенций в ИТ- сфере (Иннополис)</p>

7	<p>Знать:</p> <ul style="list-style-type: none"> – особенности объектно-ориентированной модели в Python – механизм формирования исключений – конструкции языка Python для создания потоков – примеры асинхронного программирования <p>Уметь:</p> <ul style="list-style-type: none"> – вызывать и получать результат выполнения встроенных функций Python – создать программу по описанию задачи на естественном языке с использованием функций языка Python – написать программу на Python с применением библиотек для конкретной практической задачи – использовать основные функции и методы Python для написания собственных клиентских и серверных приложений 	ПК-4
8	<p>Знать:</p> <ul style="list-style-type: none"> – принцип работы системы Git – понятия процессов и потоков <p>Уметь:</p> <ul style="list-style-type: none"> – определить, требуется ли в приложении многопоточность 	ПК-4

1.3. Категория обучающихся

Программа повышения квалификации предназначена для специалистов в области информационно-коммуникационных технологий, имеющих высшее или среднее профессиональное образование, а также лиц, получающих высшее или среднее профессиональное образование.

1.4. Форма обучения

Очная с применением дистанционных образовательных технологий.
Программа может быть реализована в сетевой форме.

1.5. Объем программы

92 академических часа.

1.6. Режим обучения

11 недель.

2. Содержание программы

2.1. Учебный (тематический) план

Учебный (тематический) план программы представлен в таблице 11.

Таблица 11

№ п/п	Наименование разделов (модулей) и тем	Всего, час.	Лекции	Практич. работа	Самост. работа	Форма контроля
УЧЕБНЫЙ ПЛАН						
1	Введение в программирование на Python	15	2,8	4	8,2	Тест, задания на программирование
2	Структуры данных и функции	16	5,9	4	6,1	Тест, задания на программирование
3	Объектно-ориентированное программирование	16	2,9	4	9,1	Тест, задания на программирование
4	Углубленный Python	24	6,9	6	11,1	Тест, задания на программирование
5	Работа с сетью	8	2,1	2	3,9	Тест, задания на программирование
6	Тестирование кода	7	1,2	2	3,8	Тест, задания на программирование
	Итоговая аттестация	6		4	2	зачет
	Итого:	92	21,8	26	44,2	
УЧЕБНЫЙ (ТЕМАТИЧЕСКИЙ) ПЛАН						
1	Введение в программирование на Python	15	2,8	4	8,2	Тест, задания на программирование
1.1.	Введение в программирование на Python	7	1,3	2	3,7	Задание на программирование
1.1. 1	Введение в программирование на Python	0,1	0,1			
1.1. 2	О языке Python	0,2	0,2			
1.1. 3	Установка интерпретатора Python на Windows	0,1	0,1			

1.1. 4	Установка интерпретатора Python на Linux	0,1	0,1			
1.1. 5	Работа в IDE PyCharm. Первая программа	0,2	0,2			
1.1. 6	Введение в Python	0,2	0,2			
1.1. 7	Ввод и вывод данных	0,2	0,2			
1.1. 8	Примеры простейших программ	0,2	0,2			
1.1. 9	Семинар	2		2		
1.1. 10	Дополнительные материалы	1,6			1,6	
1.1. 11	Тест на самопроверку по уроку	0,6			0,6	
1.1. 12	Задания на программирование	1,5			1,5	
1.2	Типы данных. Конструкции языка	8	1,5	2	4,5	Тест, задание на программирование
1.2. 1	Числовые типы и операции над ними	0,3	0,3			
1.2. 2	Строковый тип данных	0,1	0,1			
1.2. 3	Логический тип данных	0,1	0,1			
1.2. 4	Условный оператор	0,4	0,4			
1.2. 5	Цикл while	0,3	0,3			
1.2. 6	Цикл for	0,2	0,2			
1.2. 7	Операторы continue и break	0,1	0,1			
1.2. 8	Семинар	2		2		
1.2. 9	Дополнительные материалы	1			1	

1.2. 10	Тест на самопроверку по уроку	0,5			0,5	
1.2. 11	Задания на программирование	1,5			1,5	
1.2. 12	Тест на оценку	1,5			1,5	
2	Структуры данных и функции	16	5,9	4	6,1	Тест, задания на программирование
2.1	Коллекции	8	3,3	2	2,7	Задание на программирование
2.1. 1	Встроенные структуры данных	0,1	0,1			
2.1. 2	Множества. Операции с множествами	0,4	0,4			
2.1. 3	Строки. Индексация строк	0,3	0,3			
2.1. 4	Списки	0,4	0,4			
2.1. 5	Методы split() и join(). Списочные выражения	0,4	0,4			
2.1. 6	Кортежи	0,2	0,2			
2.1. 7	Словари	0,3	0,3			
2.1. 8	Примеры работы с множествами	0,2	0,2			
2.1. 9	Примеры работы со строками	0,5	0,5			
2.1. 10	Примеры работы со списками	0,3	0,3			
2.1. 11	Примеры работы с методами split() и join()	0,2	0,2			
2.1. 12	Семинар	2		2		
2.1. 13	Дополнительные материалы	0,6			0,6	
2.1. 14	Тест на самопроверку по уроку	0,6			0,6	
2.1. 15	Задания на программирование	1,5			1,5	

2.2	Функции. Работа с файлами	8	2,6	2	3,4	Тест, задание на программирование
2.2.1	Именные функции, инструкция def	0,4	0,4			
2.2.2	Возврат значений из функции	0,4	0,4			
2.2.3	Множественное присваивание, упаковка и распаковка аргументов	0,4	0,4			
2.2.4	Аргументы по умолчанию и именованные аргументы	0,2	0,2			
2.2.5	Инструкция pass(). Согласованность аргументов	0,2	0,2			
2.2.6	Функция как объект. Функции высших порядков	0,1	0,1			
2.2.7	Лямбда-функция	0,3	0,3			
2.2.8	Принципы работы с файлами на Python	0,3	0,3			
2.2.9	Разбор задач на работу с файлами	0,2	0,2			
2.2.10	Правила записи кода PEP 8	0,1	0,1			
2.2.11	Семинар	2		2		
2.2.12	Дополнительные материалы	1			1	
2.2.13	Тест на самопроверку по уроку	0,4			0,4	
2.2.14	Задания на программирование	1			1	
2.2.15	Тест на оценку	1			1	
3	Объектно-ориентированное программирование	16	2,9	4	9,1	Тест, задания на программирование

3.1.	Классы и объекты	8	1,4	2	4,6	Задание на программирование
3.1.1	Введение в объектно-ориентированное программирование	0,3	0,3			
3.1.2	Классы и экземпляры классов	0,2	0,2			
3.1.3	Методы	0,2	0,2			
3.1.4	Пример рефакторинга программы на ООП	0,4	0,4			
3.1.5	Инкапсуляция	0,2	0,2			
3.1.6	Полиморфизм	0,1	0,1			
3.1.7	Семинар	2		2		
3.1.8	Дополнительные материалы	2,5			2,5	
3.1.9	Тест на самопроверку по уроку	0,6			0,6	
3.1.10	Задания на программирование	1,5			1,5	
3.2.	Наследование	8	1,5	2	4,5	Тест, задание на программирование
3.2.1	Наследование классов	0,3	0,3			
3.2.2	Особенности объектной модели в Python	0,2	0,2			
3.2.3	Элементы статической типизации. Абстрактные классы и протоколы	0,5	0,5			
3.2.4	Множественное наследование	0,1	0,1			
3.2.5	Проблемы, связанные с наследованием	0,2	0,2			
3.2.6	Композиция классов	0,1	0,1			
3.2.7	Практические рекомендации	0,1	0,1			

3.2.8	Семинар	2		2		
3.2.9	Дополнительные материалы	1			1	
3.2.10	Тест на самопроверку по уроку	0,5			0,5	
3.2.11	Задания на программирование	1,5			1,5	
3.2.12	Тест на оценку	1,5			1,5	
4	Углубленный Python	24	6,9	6	11,1	Тест, задания на программирование
4.1	Особые методы классов. Механизм работы классов	8	1,8	2	4,2	Задание на программирование
4.1.1	Специальные методы классов	0,3	0,3			
4.1.2	Хеширование	0,4	0,4			
4.1.3	Специальные атрибуты	0,1	0,1			
4.1.4	Перегрузка операторов	0,2	0,2			
4.1.5	Коллекции и итераторы	0,3	0,3			
4.1.6	Контекстные менеджеры	0,1	0,1			
4.1.7	Callable-объекты и декораторы	0,2	0,2			
4.1.8	Построитель классов данных dataclass	0,2	0,2			
4.1.9	Семинар	2		2		
4.1.10	Дополнительные материалы	2,1			2,1	
4.1.11	Тест на самопроверку по уроку	0,6			0,6	
4.1.12	Задания на программирование	1,5			1,5	

4.2	Работа с ошибками	8	2,4	2	3,6	Задание на программирование
4.2.1	Обработка ошибок в программировании	0,3	0,3			
4.2.2	Две основные стратегии обработки ошибок	0,2	0,2			
4.2.3	Синтаксис обработки ошибок	0,4	0,4			
4.2.4	Обработка исключений и производительность	0,2	0,2			
4.2.5	Генерация исключений	0,2	0,2			
4.2.6	Инструкция assert	0,1	0,1			
4.2.7	Классы исключений	0,1	0,1			
4.2.8	Создание пользовательских исключений	0,1	0,1			
4.2.9	Практика работы с исключениями	0,8	0,8			
4.2.10	Семинар	2		2		
4.2.11	Дополнительные материалы	1,5			1,5	
4.2.12	Тест на самопроверку по уроку	0,6			0,6	
4.2.13	Задания на программирование	1,5			1,5	
4.3	Установка внешних библиотек. Работа с Git	8	2,7	2	3,3	Тест, задание на программирование
4.3.1	Инструкция import	0,2	0,2			
4.3.2	Модули стандартной библиотеки	0,4	0,4			
4.3.3	Создание своего модуля на Python	0,4	0,4			
4.3.4	Создание виртуального окружения	0,1	0,1			

4.3. 5	Установка внешних библиотек Python	0,3	0,3			
4.3. 6	Параллельная установка версий интерпретатора	0,5	0,5			
4.3. 7	Git. Работа с распределёнными системами управления версиями	0,8	0,8			
4.3. 8	Семинар	2		2		
4.3. 9	Дополнительные материалы	1			1	
4.3. 10	Тест на самопроверку по уроку	0,3			0,3	
4.3. 11	Задания на программирование	1			1	
4.3. 12	Тест на оценку	1			1	
5	Работа с сетью	8	2,1	2	3,9	Тест, задания на программирование
5.1	Работа с сетью. Сокеты	8	2,1	2	3,9	Тест, задания на программирование
5.1. 1	Сетевые протоколы	0,5	0,5			
5.1. 2	Сокеты, программа клиент-сервер	0,3	0,3			
5.1. 3	Таймауты и обработка сетевых ошибок	0,1	0,1			
5.1. 4	Одновременная обработка нескольких соединений	0,3	0,3			
5.1. 5	Простой HTTP-сервер	0,3	0,3			
5.1. 6	Основные библиотеки для работы с HTTP	0,6	0,6			
5.1. 7	Семинар	2		2		
5.1. 8	Дополнительные материалы	1,8			1,8	
5.1. 9	Тест на самопроверку по уроку	0,6			0,6	

5.1. 10	Задания на программирование	1,5			1,5	
6	Тестирование кода	7	1,2	2	3,8	Тест, задания на программирование
6.1	Контроль качества программного кода	7	1,2	2	3,8	Тест, задания на программирование
6.1. 1	Обеспечение качества и тестирование ПО	0,2	0,2			
6.1. 2	Инструменты статического анализа кода	0,3	0,3			
6.1. 3	Инструменты тестирования	0,2	0,2			
6.1. 4	Использование фикстур и мок-объектов в pytest	0,4	0,4			
6.1. 5	Покрытие кода тестами. Плагин pytest-cov	0,1	0,1			
6.1. 6	Семинар	2		2		
6.1. 7	Дополнительные материалы	1,7			1,7	
6.1. 8	Тест на самопроверку по уроку	0,6			0,6	
6.1. 9	Задания на программирование	1,5			1,5	
7	Итоговая аттестация	6		4	2	Зачет
	Итого:	92	21,8	26	44,2	

2.2. Учебная программа

Содержание учебной программы приведено в таблице 12.

Таблица 12

№ п/п	Наименование модуля, разделов и тем	Содержание обучения, наименование и тематика практических занятий (семинаров), самостоятельной работы	Объем, ак. час.
1	Введение в программирование на Python		15
1.1	Введение в программирование на Python	<i>Лекция</i> Вводное видео к курсу Введение в программирование на Python О языке Python Установка интерпретатора Python на Windows Установка интерпретатора Python на Linux Работа в IDE PyCharm. Первая программа Введение в Python Ввод и вывод данных Примеры простейших программ Итоги занятия <i>Практическая работа</i> Выполнение заданий по теме лекции. <i>Самостоятельная работа</i> Дополнительные материалы Тест на самопроверку по уроку Задания на программирование	7

1.2	Типы данных. Конструкции языка	<p>Лекция</p> <p>Числовые типы и операции над ними Строковый тип данных Логический тип данных Условный оператор Цикл while Цикл for Операторы continue и break</p> <p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Дополнительные материалы Тест на самопроверку по уроку Задания на программирование Тест на оценку</p>	8
2	Структуры данных и функции		16
2.1	Коллекции	<p>Лекция</p> <p>Встроенные структуры данных Множества. Операции с множествами Строки. Индексация строк Списки Методы split() и join(). Списочные выражения Кортежи Словари Примеры работы с множествами Примеры работы со строками Примеры работы со списками Примеры работы с методами split() и join()</p> <p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Дополнительные материалы Тест на самопроверку по уроку Задания на программирование</p>	8

2.2	Функции. Работа с файлами	<p>Лекция</p> <p>Именные функции, инструкция def Возврат значений из функции Множественное присваивание, упаковка и распаковка аргументов Аргументы по умолчанию и именованные аргументы Инструкция pass(). Согласованность аргументов Функция как объект. Функции высших порядков Лямбда-функция Принципы работы с файлами на Python Разбор задач на работу с файлами Правила записи кода PEP 8</p> <p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Дополнительные материалы Тест на самопроверку по уроку Задания на программирование Тест на оценку</p>	8
3	Объектно-ориентированное программирование	16	
3.1	Классы и объекты	<p>Лекция</p> <p>Введение в объектно-ориентированное программирование Классы и экземпляры классов Методы Пример рефакторинга программы на ООП Инкапсуляция Полиморфизм</p> <p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Дополнительные материалы Тест на самопроверку по уроку Задания на программирование</p>	8

3.2	Наследование	<p>Лекция</p> <p>Наследование классов Особенности объектной модели в Python Элементы статической типизации. Абстрактные классы и протоколы Множественное наследование Проблемы, связанные с наследованием Композиция классов Практические рекомендации</p> <p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Дополнительные материалы Тест на самопроверку по уроку Задания на программирование Тест на оценку</p>	8
4	Углубленный Python		24
4.1	Особые методы классов. Механизм работы классов	<p>Лекция</p> <p>Специальные методы классов Хеширование Специальные атрибуты Перегрузка операторов Коллекции и итераторы Контекстные менеджеры Callable-объекты и декораторы Построитель классов данных dataclass</p> <p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Дополнительные материалы Тест на самопроверку по уроку Задания на программирование</p>	8

4.2	Работа с ошибками	<p>Лекция</p> <p>Обработка ошибок в программировании Две основные стратегии обработки ошибок Синтаксис обработки ошибок Обработка исключений и производительность Генерация исключений Инструкция assert Классы исключений Создание пользовательских исключений Практика работы с исключениями</p> <p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Дополнительные материалы Тест на самопроверку по уроку Задания на программирование</p>	8
4.3	Установка внешних библиотек. Работа с Git	<p>Лекция</p> <p>Инструкция import Модули стандартной библиотеки Создание своего модуля на Python Создание виртуального окружения Установка внешних библиотек Python Параллельная установка версий интерпретатора Git. Работа с распределёнными системами управления версиями</p> <p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Тест на самопроверку по уроку Задания на программирование Тест на оценку</p>	8
5	Работа с сетью		8

5.1	Работа с сетью. Сокеты	<p>Лекция</p> <p>Сетевые протоколы Сокеты, программа клиент-сервер Таймауты и обработка сетевых ошибок Одновременная обработка нескольких соединений Простой HTTP-сервер Основные библиотеки для работы с HTTP</p> <p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Дополнительные материалы Тест на самопроверку по уроку Задания на программирование</p>	8
6	Тестирование кода		7
6.1	Контроль качества программного кода	<p>Лекция</p> <p>Обеспечение качества и тестирование ПО Инструменты статического анализа кода Инструменты тестирования Использование фикстур и мок-объектов в pytest Покрытие кода тестами. Плагин pytest-cov</p> <p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Дополнительные материалы Тест на самопроверку по уроку Задания на программирование</p>	7
	Итоговая аттестация	зачет	6
	Итого:		92

3. Формы аттестации и оценочные материалы

3.1. Формы аттестации

Итоговая аттестация по программе является обязательной.

Форма итоговой аттестации предусматривает зачет в форме тестирования.

Оценка «зачтено» выставляется обучающемуся, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка «не зачтено» выставляется обучающемуся, 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.

Максимальная сумма, которую можно набрать, успешно выполнив все контрольные мероприятия, составляет 100 баллов. Для получения положительной оценки «зачтено» необходимо набрать не менее 30 баллов.

В зачетно-экзаменационную ведомость оценка выставляется в соответствии с нижеприведенной таблицей 13.

Таблица 13

Сумма баллов	Оценка
30-100	Зачтено
менее 30	Не зачтено

Составляющие процесса обучения, которые оцениваются в ходе обучения, и их вклад в итоговую оценку представлены в таблице 14.

Таблица 14

№ п/п	Основные показатели оценки	Вклад в итоговую оценку
1	Практические задания	30%
2	Выполнение домашних заданий	40%
3	Промежуточная (итоговая) аттестация	30%

3.2.Оценочные материалы

Таблица 15

Наименование модуля, разделов и тем	Основные показатели оценки	Формы и методы контроля и оценки	Вес задания, %
Введение в программирование на Python	ПК-1, ПК-3, ПК-4	Тест, задание на программирование	15
Структуры данных и функции	ОПК-8, ОПК-9, ПК-1	Тест, задание на программирование	15
Объектно-ориентированное программирование	ПК-1, ПК-3, ПК-4	Тест, задание на программирование	15
Углубленный Python	ОПК-8, ОПК-9, ПК-2, ПК-5	Тест, задание на программирование	15
Работа с сетью	ПК-1, ПК-3, ПК-4	Тест, задание на программирование	15
Тестирование кода	ПК-1, ПК-3, ПК-4	Тест, задание на программирование	15
Итоговая аттестация	ОПК-8, ОПК-9, ПК-1, ПК-2, ПК-3, ПК-4, ПК-5	Тест	10

Примеры заданий

Тестовые задания

1. Какая версия языка Python является актуальной в настоящее время?
 - Python 3.X
 - Python 2.6.X
 - Python 1.X
2. Что такое инструкция в языках программирования?
 - команда для выполнения какого-либо действия
 - документ с перечислением всех команд языка
 - данные, которые пользователь вводит с консоли в программу
 - спецсимволы, используемые для написания кода

3. Какие элементы используют для формирования структуры программы в Python? Выберите все верные варианты.

- пробелы
- двоеточие
- табуляция
- фигурные скобки
- команды языка
- знак #

4. Людмила пишет программу на языке Python. Какую команду ей нужно записать на место пропуска, что вывести на экран сумму чисел a и b? Запишите ответ строчными буквами.

Код программы Людмилы.

a = 2

b = 4

Практические задания

Задание на программирование

Цель заданий

В результате выполнения заданий вы сможете:

- разобраться с особенностями написания кода в LMS
- писать программы на Python с использованием ввода/вывода в консоль

Задания:

1. Приветствие

На вход программа получает строку — имя пользователя. Считайте её и выведите на экран фразу: «Привет, [имя]!».

Обратите внимание, что выводимая строка должна быть написана в точности так, как в задании. Если будет отличаться регистр букв или будут отсутствовать знаки препинания, программа засчитана не будет.

Ввод	Вывод
Валентин	Привет, Валентин!

Авторское решение:

```
name = input()
print('Привет, ', name, '!', sep="")
```

Тесты

№ теста	Стандартный ввод	Ожидаемый результат
1	Валентин	Привет, Валентин!
2	Дмитрий	Привет, Дмитрий!
3	Мария	Привет, Мария!
4	АБВГД123	Привет, АБВГД123!

2. Вычисление значения предыдущего числа и следующего

На вход программа получает целое число a . Получите значения предыдущего числа и следующего. Выведите результат $a-1$, a , $a+1$ по одному числу в строку.

Ввод	Результат
5	4 5 6

Авторское решение

```
a = int(input())
print(a - 1)
print(a)
print(a + 1)
```

Тесты

Тест 1:

ввод	Ожидаемый результат
5	4 5 6

Тест 2:

ввод	Ожидаемый результат
0	-1 0 1

Авторское решение

```
name = input()
animal = input()
breed = input()
age = input()
print('Ваш питомец:', animal, name)
print('Порода питомца:', breed)
print('Возраст питомца:', age)
```

Тест 1:

Ввод	Вывод
Кузя кот мейн-кун 5	Ваш питомец: кот Кузя Порода питомца: мейн-кун Возраст питомца: 5

Тест 2:

Ввод	Вывод
Маруся кошка британская длинношерстная 4	Ваш питомец: кошка Маруся Порода питомца: британская длинношерстная Возраст питомца: 4

Тест 3:

Ввод	Вывод
Бонни собака бульдог 8	Ваш питомец: собака Бонни Порода питомца: бульдог Возраст питомца: 8

4. Организационно-педагогические условия реализации программы

4.1. Учебно-методическое обеспечение и информационное обеспечение программы

4.1.1. Список литературы

Основная литература

1. Think Python [Электронный ресурс] – Режим доступа - <https://greenteapress.com/wp/think-python-2e/>
2. Automate the Boring Stuff with Python [Электронный ресурс] – Режим доступа - <https://automatetheboringstuff.com/>
3. Dive Into Python 3 [Электронный ресурс] – Режим доступа - <http://diveintopython3.problemsolving.io/>
4. Problem Solving with Algorithms and Data Structures using Python [Электронный ресурс] – Режим доступа - <https://runestone.academy/runestone/static/pythonds/index.html>
5. Swaroop Chitlur. A Byte of Python [Электронный ресурс] – Режим доступа - <https://wombat.org.ua/AByteOfPython/AByteofPythonRussian-2.02.pdf> – 2020.
6. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для вузов — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2021. — 210 с
7. Свейгарт Э. Автоматизация рутинных задач с помощью Python [Текст]: практическое руководство для начинающих – Диалектика, 2021. – 672 с.
8. Книга издательства O'Reilly (доступна онлайн, англ.): <https://docs.python-guide.org/>
9. Лутц, Марк. Изучаем Python: авторитетный курс объектно-ориентированного программирования / Марк Лутц; перевод с английского Ю. Н. Артеменко. – 5-е изд. – Москва: Диалектика; Санкт-Петербург: Диалектика, 2020.
10. Учебная статья: <https://docs.python.org/3/tutorial/errors.html>

Дополнительная литература

1. Бэрри П. Изучаем программирование на Python [Текст] / П. Бэрри. – 2017.
2. Маккинни У. Python и анализ данных [Текст] / У. Маккинни. – 2015.

4.1.2. Ресурсы информационно-телекоммуникационной сети "Интернет"

1. Официальный сайт Python - <https://www.python.org/>
2. Введение в анализ данных с помощью Pandas - <https://habr.com/ru/post/196980/>
3. Среда разработки IDE PyCharm - <https://www.jetbrains.com/pycharm/>
4. PEP 8 – Style Guide for Python Code - <https://peps.python.org/pep-0008/>
5. Официальная документация (англ.): <https://docs.python.org/3/>
6. Документация по исключениям: <https://docs.python.org/3/library/exceptions.html>

7. Модель GitFlow: <https://nvie.com/posts/a-successful-git-branching-model/>
8. Современный подход к версионированию для CI/CD: <https://www.atlassian.com/continuous-delivery/continuous-integration/trunk-based-development>

4.2. Описание материально-технической базы, необходимой для осуществления образовательного процесса по программе

Таблица 16

Наименование специализированных аудиторий, кабинетов, лабораторий	Вид занятий	Наименование оборудования, программного обеспечения
Система дистанционного обучения	Лекция	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю курса необходимо наличие доступа администратора курса и оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.
Система дистанционного обучения	Практическая работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер.
Система дистанционного обучения	Самостоятельная работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер.
Система дистанционного обучения	Промежуточная (итоговая) аттестация	Слушателю необходимо наличие доступа в сеть интернет, компьютер.

5. Организация образовательного процесса

В таблице 17 описаны образовательные технологии.

Таблица 17

№ п/п	Вид занятия	Форма проведения занятий	Цель
1	Лекция	Самостоятельный просмотр видеолекций Обсуждение вопросов, возникших в результате просмотра видеолекций, выполнения практических заданий и изучения литературы.	актуализация и систематизация теоретических знаний по дисциплине
2	Практическая работа	Выполнение практических заданий. Прохождение практики в организации партнера в дистанционном формате.	осознание связей между теорией и практикой, повышение степени понимания материала
3	Самостоятельная работа	Самостоятельное изучение дополнительных материалов и литературы.	получение дополнительных теоретических знаний
4	Промежуточная (итоговая) аттестация	Выполнение заданий.	контроль освоения программы

РАБОЧАЯ ПРОГРАММА МОДУЛЯ 2

Объектно-ориентированное программирование, графический интерфейс и основы работы с базами данных в Python

1. Цель освоения Модуля 2

Целью реализации модуля «Объектно-ориентированное программирование, графический интерфейс и основы работы с базами данных в Python» является совершенствование компетенций слушателей в области разработки простых программ и приложений на языке Python с использованием графического интерфейса (GUI), а также основ работы с базами данных в Python.

1.1. Совершенствуемые и/или приобретаемые компетенции

Компетенции, формируемые и совершенствуемые в результате обучения, представлены в таблицах 18 и 19.

Таблица 18

№	Компетенции в соответствии с профессиональным стандартом 06.001 Программист	Код компетенции
1	Способен написать программный код с использованием языков программирования, определения и манипулирования данными	ПК-1
2	Способен работать с системой контроля версий	ПК-2
3	Способен проверить и отладить программный код	ПК-5

Таблица 19а

№	Компетенции в соответствии с направлением подготовки 09.03.01 Информатика и вычислительная техника (ФГОС)	Код компетенции
1.	Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8
2	Способен осваивать методики использования программных средств для решения практических задач	ОПК-9

Таблица 19б

№	Профессиональные компетенции (ПК) в соответствии с направлением подготовки 09.03.01 Информатика и вычислительная техника (Собственный стандарт МФТИ)	Код компетенции
1.	Способен формализовать и алгоритмизировать поставленную задачу, написать программный код с использованием языков программирования, оформить код в соответствии с установленными требованиями	ПК-3

Профессиональные компетенции (ПК) в соответствии с Матрицей (моделью) цифровых компетенций в части профессиональных компетенций в ИТ-сфере (Иннополис)

Таблица 19в

Наименование сферы	Код и наименование профессиональной компетенции	БАЗОВЫЙ УРОВЕНЬ РАЗВИТИЯ КОМПЕТЕНЦИЙ Способность проявляется под внешним контролем / при внешней постановке задачи/ обучающийся пользуется готовыми, рекомендованными продуктами
Средства программной разработки	ПК-4 Применяет языки программирования для решения профессиональных задач	Самостоятельно применяет языки программирования. Использует настраиваемые программные инструменты для автоматизации процессов в профессиональной деятельности.
Средства программной разработки	ПК-7 Применяет СУБД	Знает основы баз данных, знаком с нормализацией, АСІD, транзакциями, может написать простые выборки. Участвует в проектах по созданию ПО с использованием СУБД под контролем опытных специалистов

1.2. Планируемые результаты обучения

Планируемые результаты обучения представлены в таблице 20.

Таблица 20

№	Уметь - знать	Направление подготовки 09.03.01 Информатика и вычислительная техника (ФГОС)
		Квалификация: бакалавр
		Код компетенции
1	Знать: <ul style="list-style-type: none"> – первичных ключей, типов данных атрибутов умеет строить дерево решений для выбора СУБД – принципы проектирования SOLID – смысл принципов проектирования SOLID – правильную структуру документа Уметь: <ul style="list-style-type: none"> – типы структур данных, операций над данными – составлять описание требований к будущему программному обеспечению 	ОПК-8
2	Знать:	ОПК-9

	<ul style="list-style-type: none"> – принципы проектирования БД, применяемые в работе программ и приложений <p>Уметь:</p> <ul style="list-style-type: none"> – запускать анимации – работать с кривой плавности – делать анимацию для нескольких объектов 	
		Профессиональный стандарт 06.001 Программист
		Код компетенции
3	<p>Знать:</p> <ul style="list-style-type: none"> – основные паттерны ООП – в каких ситуациях применять основные паттерны ООП <p>Уметь:</p> <ul style="list-style-type: none"> – таблицу в БД, создавать БД через консоль, удалять таблицу из БД, выгружать БД в SQL формате, отображать все таблицы в БД, отображать структуру БД – рисовать средствами QPainter. – рисовать средствами QCanvas. 	ПК-1
4	<p>Знать:</p> <ul style="list-style-type: none"> – термины нормализации – Основные нормальные формы: 1НФ, 2НФ, 3НФ, НФБК Прочие нормальные формы: 4NF, 5NF, DKNF, 6NF <p>Уметь:</p> <ul style="list-style-type: none"> – добавлять записи – использовать паттерны, стандартизировать код 	ПК-2
5	<p>Уметь:</p> <ul style="list-style-type: none"> – обращаться к записям, фильтровать поиск – вносить изменения в записи, удалять записи – устанавливать соединения и выполнение запроса. – тестировать код БД, проверять на ошибки, работать с исключениями 	ПК-5
		Направление подготовки 09.03.01 Информатика и вычислительная техника (Собственный стандарт МФТИ)
6	<p>Знать:</p> <ul style="list-style-type: none"> – как должно выглядеть представление данных в табличной форме. – свойства таблицы стилей QSS и умеет их применять <p>Уметь:</p>	ПК-3

	<ul style="list-style-type: none"> – Проектировать системы с использованием паттернов – разрабатывать формы, ориентированные на работу с базами данных. – создавать потоки и ориентироваться в них. – настраивать взаимодействие с главным потоком приложения. – управлять циклом внутри потока. – создавать окно, элементы в нем. – перемещать элементы внутри окна. – добавлять элементам цвет. 	
		Матрица (модель) цифровых компетенций в части профессиональных компетенций в ИТ-сфере (Иннополис)
7	<p>Знать:</p> <ul style="list-style-type: none"> – синтаксис Python – углубленно среду для разработки, ее особенности, может использовать ее для своей работы – виджеты и сигналы <p>Уметь:</p> <ul style="list-style-type: none"> – выполнять сложные запросы SELECT, объединять таблицы, делать подзапросы и группировать таблицы и БД – оптимизировать однотипные виджеты – создавать окна в PyQt – работать с классами Qt вне главного потока. – изменять существующие виджеты Qt и кастомизировать их. 	ПК-4
8	<p>Знать:</p> <ul style="list-style-type: none"> – определение и основные характеристики БД – отличия БД от таблицы в Excel – свойства БД (быстродействие, быстрота, независимость, стандартизация, безопасность, интегрированность, многопользовательский доступ) – типы БД и их отличия друг от друга (иерархические, сетевые, объектно-ориентированные, реляционные, нереляционные) – что такое реляционная база данных и СУБД Реляционная модель данных – Основные функции реляционной СУБД – Особенности проектирования реляционных БД <p>Уметь:</p> <ul style="list-style-type: none"> – перечислить функции БД 	ПК-7

	<ul style="list-style-type: none"> – привести примеры использования БД – подключаться к БД, создавать/ удалять БД и таблицы, заполнять/ редактировать/ выводить таблицы, импортировать и экспортировать файлы, PRIMARY KEY и FOREIGN KEY – строить БД под требования приложения 	
--	--	--

1.3. Категория обучающихся

Программа повышения квалификации предназначена для лиц, имеющих высшее образование и желающих получить компетенции в сфере программирования на Python для своей профессиональной деятельности; студентов вузов.

1.4. Форма обучения

Очная с применением дистанционных образовательных технологий.

Программа может быть реализована в сетевой форме.

1.5. Объем программы

61 академический час.

1.6. Режим обучения

7 недель.

2. Содержание программы

2.1. Учебный (тематический) план

Учебный (тематический) план программы представлен в таблице 21.

Таблица 21

№ п/п	Наименование разделов (модулей) и тем	Всего, час.	Лекции	Практич. работа	Самост. работа	Форма контроля
УЧЕБНЫЙ ПЛАН						
1	Работа с базами данных	15	4,3	4	6,7	Тест, задание на программирование
2	Паттерны ООП на Python для разработки приложения	8	2,1	2	3,9	Тест, задание на программирование

3	Построение графических интерфейсов	32	8,7	8	15,3	Задание проекта
	Итоговая аттестация	6		4	2	зачет
	Итого:	61	15,1	18	25,9	
УЧЕБНЫЙ (ТЕМАТИЧЕСКИЙ) ПЛАН						
1	Работа с базами данных	15	4,3	4	6,7	Тест, задание на программирование
1.1	Введение в базы данных. Знакомство с SQL	7	2,1	2	2,9	Тест
1.1.1	Введение в базы данных 1	0,6	0,6			
1.1.2	Реляционные базы данных СУБД 2	0,3	0,3			
1.1.3	SQL. Создание базы данных 3	0,3	0,3			
1.1.4	Основные операции с таблицами 4	0,9	0,9			
1.1.5	Семинар 5	2		2		
1.1.6	Дополнительные материалы 6	1,9			1,9	
1.1.7	Тест на самопроверку по уроку 7	1			1	Тест
1.2	Проектирование базы данных	8	2,2	2	3,8	Тест, задание на программирование

1.2. 1	Сложные запросы SELECT, объединение таблиц, подзапросы, группировка	0,9	0,9			
1.2. 2	Функции SQL	0,6	0,6			
1.2. 3	Нормализация и тестирование БД	0,4	0,4			
1.2. 4	Принципы проектирования БД	0,1	0,1			
1.2. 5	Модуль sqlite3 языка Python	0,2	0,2			
1.2. 6	Семинар	2		2		
1.2. 7	Дополнительные материалы	0,5			0,5	
1.2. 8	Тест на самопроверку по уроку	0,8			0,8	Тест
1.2. 9	Задания на программирование	1			1	Задания на программирова ние
1.2. 10	Тест на оценку	1,5			1,5	Тест
2	Паттерны ООП на Python для разработки приложения	8	2,1	2	3,9	Тест, задание на программирова ние
2.1	Качество кода	0,4	0,4			
2.2	SOLID-принципы	0,8	0,8			
2.3	Порождающие паттерны проектирования	0,3	0,3			

2.4	Структурные паттерны	0,3	0,3			
2.5	Поведенческие паттерны	0,3	0,3			
2.6	Семинар	2		2		
2.7	Дополнительные материалы	0,4			0,4	
2.8	Тест на самопроверку по уроку	0,5			0,5	Тест
	Задания на программирование	1,5			1,5	Задания на программирование
	Тест на оценку	1,5			1,5	Тест
3	Построение графических интерфейсов	32	8,7	8	15,3	Задание проекта
3.1	Этап I. Учимся проектировать приложение	8	2,4	2	3,6	Задание проекта
3.1.1	Разработка ТЗ	0,4	0,4			
3.1.2	Архитектура проекта	0,6	0,6			
3.1.3	Средства разработки	0,4	0,4			
3.1.4	Структура проекта	0,2	0,2			
3.1.5	Пишем репозиторий	0,6	0,6			
3.1.6	Пишем модели	0,2	0,2			
3.1.7	Семинар	2		2		

3.1. 8	Дополнительные материалы	0,6			0,6	
3.1. 9	Задание проекта	3			3	Задание проекта
3.2	Этап II. Основы графического интерфейса приложения	8	1,8	2	4,2	Задание проекта
3.2. 1	Введение в программирование GUI	0,3	0,3			
3.2. 2	Основы PyQt.	0,4	0,4			
3.2. 3	Виджеты и сигналы.	0,5	0,5			
3.2. 4	Раскладки (layouts)	0,2	0,2			
3.2. 5	Программирование виджетов в ООП стиле	0,4	0,4			
3.2. 6	Семинар	2		2		
3.2. 7	Дополнительные материалы	1,2			1,2	
3.2. 8	Задание проекта	3			3	Задание проекта
3.3	Этап III. Продвинутое возможности библиотеки PySide	8	2	2	4	Задание проекта
3.3. 1	Окна в PyQt	0,4	0,4			
3.3. 2	Сигналы и события	0,5	0,5			

3.3. 3	Многопоточное программирование в PyQt	0,3	0,3			
3.3. 4	Тестирование PyQt-приложений	0,4	0,4			
3.3. 5	Интегрирование GUI в архитектуру MVP	0,4	0,4			
3.3. 6	Семинар	2		2		
3.3. 7	Дополнительные материалы	1			1	
3.3. 8	Задание проекта	3			3	Задание проекта
3.4	Этап IV. Кастомизация интерфейса приложения	8	2,5	2	3,5	Задание проекта
3.4. 1	Кастомизация окна	0,5	0,5			
3.4. 2	Создание виджетов	0,3	0,3			
3.4. 3	Определение стилей	1	1			
3.4. 4	Анимации	0,7	0,7			
3.4. 5	Семинар	2		2		
3.4. 6	Дополнительные материалы	0,5			0,5	
3.4. 7	Задание проекта	3			3	Задание проекта
	Итоговая аттестация	6		4	2	зачет
	Итого:	61	15,1	18	25,9	

2.2. Учебная программа

Содержание учебной программы приведено в таблице 22.

Таблица 22

№ п/п	Наименование модуля, разделов и тем	Содержание обучения, наименование и тематика практических занятий (семинаров), самостоятельной работы	Объем, ак.час.
1	Работа с базами данных		15
1.1	Введение в базы данных. Знакомство с SQL	Лекция Введение в базы данных Реляционные базы данных СУБД SQL. Создание базы данных Основные операции с таблицами Практическая работа Выполнение заданий по теме лекции Самостоятельная работа Дополнительные материалы Данные для практической работы supermarket_sales Тест на самопроверку по уроку	7
1.2	Проектирование базы данных	Лекция Сложные запросы SELECT, объединение таблиц, подзапросы, группировка Функции SQL Нормализация и тестирование БД Принципы проектирования БД Модуль sqlite3 языка Python Практическая работа Задание на работу с БД Самостоятельная работа Дополнительные материалы Тест на самопроверку по уроку Задания на программирование Тест на оценку	8
2	Паттерны ООП на Python для разработки приложения	Лекция Качество кода SOLID-принципы Порождающие паттерны проектирования Структурные паттерны Поведенческие паттерны Практическая работа Выполнение заданий по теме лекции Самостоятельная работа	8

		Дополнительные материалы Тест на самопроверку по уроку Задания на программирование Тест на оценку	
3	Построение графических интерфейсов		32
3.1	Этап I. Учимся проектировать приложение	Лекция Разработка ТЗ Архитектура проекта Средства разработки Структура проекта Пишем репозиторий Пишем модели Практическая работа Выполнение заданий по теме лекции Самостоятельная работа Семинар Дополнительные материалы Задание проекта	8
3.2	Этап II. Основы графического интерфейса приложения	Лекция Введение в программирование GUI Основы PyQt Виджеты и сигналы Раскладки (layouts) Программирование виджетов в ООП стиле Практическая работа Выполнение заданий по теме лекции Самостоятельная работа Дополнительные материалы Задание проекта	8
3.3	Этап III. Продвинутое возможности библиотеки PySide	Лекция Окна в PyQt Сигналы и события Многопоточное программирование в PyQt Тестирование PyQt-приложений Интегрирование GUI в архитектуру MVP Практическая работа Выполнение заданий по теме лекции Самостоятельная работа Дополнительные материалы Задание проекта	8
3.4	Этап IV. Кастомизация интерфейса приложения	Лекция Кастомизация окна Создание виджетов Определение стилей Анимации Практическая работа	8

		Выполнение заданий по теме лекции <i>Самостоятельная работа</i> Дополнительные материалы Задание проекта	
	Итоговая аттестация	зачет	6
	Итого:		61

3. Формы аттестации и оценочные материалы

3.1. Формы аттестации

Итоговая аттестация по программе является обязательной.

Форма итоговой аттестации предусматривает зачет в форме тестирования.

Оценка «зачтено» выставляется обучающемуся, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка «не зачтено» выставляется обучающемуся, 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.

Максимальная сумма, которую можно набрать, успешно выполнив все контрольные мероприятия, составляет 100 баллов. Для получения положительной оценки «зачтено» необходимо набрать не менее 30 баллов.

В зачетно-экзаменационную ведомость оценка выставляется в соответствии с нижеприведенной таблицей 23.

Таблица 23

Сумма баллов	Оценка
30-100	Зачтено
менее 30	Не зачтено

Составляющие процесса обучения, которые оцениваются в ходе обучения, и их вклад в итоговую оценку представлены в таблице 24.

Таблица 24

№ п/п	Основные показатели оценки	Вклад в итоговую оценку
1	Практические задания	30%
2	Выполнение домашних заданий	40%
3	Промежуточная (итоговая) аттестация	30%

3.2.Оценочные материалы

Таблица 25

Наименование модуля, разделов и тем	Основные показатели оценки	Формы и методы контроля и оценки	Вес задания, %
Работа с базами данных	ОПК-8, ОПК-9, ПК-3, ПК-1, ПК-2, ПК-5, ПК-7	Тест, задание на программирование	20
Паттерны ООП на Python для разработки приложения	ОПК-9, ПК-3, ПК-1, ПК-2, ПК-5, ПК-7	Тест, задание на программирование	20
Построение графических интерфейсов	ОПК-9, ПК-3, ПК-1, ПК-2, ПК-5, ПК-4	Задание проекта	20
Итоговая аттестация	ОПК-8, ОПК-9, ПК-1, ПК-2, ПК-3, ПК-4, ПК-5, ПК-7	Задание проекта	40

Примеры заданий

Тестовые задания

1. Есть большой объём данных по клиентам. В ходе работы необходимо обращаться к небольшому числу столбцов. При этом критична скорость работы. Какой тип модели данных лучше выбрать?
 - Колоночные (столбчатые) базы данных (*Clickhouse, Vertica*)
 - Реляционные базы данных (*SQLite, Postgres*)
 - Графовые базы данных
 - Иерархические базы данных
2. Данные, которые нужно хранить в базе данных представимы в виде набора связанных таблиц. Какой тип базы данных будет подходящим?
 - Реляционная БД
 - База данных с широкими столбцами
 - Графовая база данных
 - Иерархическая
3. В таблице с информацией о клиентах есть поля возраст клиента, адрес клиента, номер телефона, дата регистрации. Какое из полей может использоваться в качестве первичного ключа?
 - номер телефона
 - возраст клиента
 - дата регистрации
 - адрес клиента

4. В таблице с информацией о недвижимости (estate) есть поле с этажом квартир (floor). Нужно выбрать строки с нечётным этажом. Как это можно сделать?
- `SELECT * FROM estate WHERE floor % 2 != 0;`
 - `SELECT * FROM estate;`
 - `SELECT * FROM estate WHERE floor is ODD;`
 - `SELECT * FROM estate WHERE floor // 2 != 0;`

Задание на программирование

- А) Создайте базу данных (БД), используя программу DB Browser for SQLite, с названием **Python_<фамилия на латинице>**.

Б) Добавьте в созданную БД:

 - 1) Таблицу **Users** с полями: уникальный идентификатор пользователя, имя, фамилия, возраст, город проживания, страна, дата регистрации в онлайн-магазине.
 - 2) Таблицу **Products** с полями: уникальный идентификатор товара, название товара, категория товара, стоимость товара за ед. в \$.
 - 3) Таблицу **Sales** с полями: уникальный идентификатор транзакции, дата совершения транзакции, уникальный идентификатор купленного товара, количество ед. товара, наличие скидки, пункт выдачи/адрес доставки товара

В) Заполните таблицы выдуманными данными (по 5-10 строк).

Г) Перейдите во вкладку SQL и напишите скрипты, чтобы выполнить следующие действия с БД:

 - 1) В таблице **Users** присвойте всем пользователям город проживания Москва.
 - 2) В таблице **Users** измените возраст клиента во 2-й строке таблицы на 35 лет.
 - 3) В таблице **Products** добавьте новый столбец со стоимостью товара в рублях (курс можно взять с сайта ЦБ на дату выполнения ДЗ)
 - 4) В таблице **Sales** выберите строки, для которых дата совершения транзакции наиболее ранняя или наиболее поздняя, т. е. строки с минимальной и максимальной датами.
- Загрузите данные Supermarket_sales и импортируйте их в DB Browser for SQLite с названием **SupermarketSales_<фамилия на латинице>**. По ссылке можно посмотреть описание базы данных <https://www.kaggle.com/datasets/aungpyaeap/supermarket-sales>

Перейдите во вкладку SQL и напишите скрипты, чтобы выполнить следующие действия с БД:

 - 1) Выберите записи таблицы, в которых поле "**Branch**" принимает значение А.
 - 2) Выберите записи таблицы, в которых поле "**Branch**" принимает значение А или С.
 - 3) Выберите записи таблицы, в которых поле "**Branch**" принимает значение отличное от А.
 - 4) Выберите записи таблицы, в которых поле "**Branch**" принимает значение В, а поле **Productline** равно "**Fashion accessories**".

- 5) Удалите записи таблицы, для которых поле **“Rating”** принимает значение меньше 5.
3. С помощью SQL запросов к БД из задания 2 ответьте на вопросы:
- 1) Какое количество транзакций совершили клиенты?

Одна строка соответствует одной транзакции. После удаления строк (п.5 задания 2) у нас остается 847 строк

- 2) Сколько транзакций с товарами категории **«Electronic accessories»** было совершено?

Запрос должен выглядеть так: `SELECT * FROM sales WHERE Productline == "Electronic accessories"`

- 3) Были ли покупки товара категории **«Electronic accessories»** за период с 3 февраля 2019 года? Если такие покупки были, то найдите их число.

Запросы должны выглядеть так:

```
SELECT InvoiceID, strftime('%d.%m.%Y', Date) as trans_date
FROM supermarket_sales_data
WHERE (trans_date >= date('2019-02-03')) & (Productline = 'Electronic
accessories');
```

- 4) Клиентов с каким рейтингом больше
- а) 5-7
- б) больше 8?

Запросы должны выглядеть так:

```
SELECT * FROM sales WHERE Rating == 5 OR Rating == 6 OR Rating == 7;
SELECT * FROM sales WHERE Rating > 8;
```

- 5) Сколько типов оплаты (поле **Payment**)?

Запрос должен выглядеть так: `SELECT DISTINCT (Payment) FROM sales;`

- 6) Найдите стоимости (**Unitprice**) и категории (**Productline**) для товаров, купленных в количестве (**Quantity**) 10 шт. за одну транзакцию. Напишите число таких транзакций.

Запрос должен выглядеть так: `SELECT Unitprice, Productline FROM sales WHERE Quantity = 10;`

4. Организационно-педагогические условия реализации программы

4.1. Учебно-методическое обеспечение и информационное обеспечение программы

4.1.1. Список литературы

Основная литература

1. Think Python [Электронный ресурс] – Режим доступа - <https://greenteapress.com/wp/think-python-2e/>
2. Automate the Boring Stuff with Python [Электронный ресурс] – Режим доступа - <https://automatetheboringstuff.com/>
3. Dive Into Python 3 [Электронный ресурс] – Режим доступа - <http://diveintopython3.problemsolving.io/>
4. Problem Solving with Algorithms and Data Structures using Python [Электронный ресурс] – Режим доступа - <https://runestone.academy/runestone/static/pythonds/index.html>
5. Swaroop Chitlur. A Byte of Python [Электронный ресурс] – Режим доступа - <https://wombat.org.ua/AByteOfPython/AByteofPythonRussian-2.02.pdf> – 2020.
6. Федоров Д. Основы программирования на примере языка Python [Текст] : учебное пособие / Д. Федоров. - 2018.
7. Свейгарт Э. Автоматизация рутинных задач с помощью Python [Текст] : практическое руководство для начинающих / Эл Свейгарт. – Диалектика, 2021.
8. Роберт Мартин. Чистый код: создание, анализ и рефакторинг.
9. Марк Саммерфилд. Python на практике.
10. Гарри Персиваль, Боб Грегори. Паттерны разработки на Python.
11. Прохоренок Н.А., Дронов В.А. Python 3 и PyQt6. Разработка приложений. – СПб.: БХВ-Петербург, 2023
12. Harry J.W. Percival. "Test-Driven Web Development with Python": <http://www.obeythetestinggoat.com/pages/book.html#toc>

Дополнительная литература

1. Бэрри П. Изучаем программирование на Python [Текст] / П. Бэрри. – 2017.

4.1.2. Ресурсы информационно-телекоммуникационной сети "Интернет"

1. PySide6.QtWidgets - <https://doc.qt.io/qtforpython/PySide6/QtWidgets/index.html>
2. Документация Postgres про сравнение строк - <https://postgrespro.ru/docs/postgrespro/9.5/functions-matching>
3. Документация Postgres про другие функции работы со строками - <https://postgrespro.ru/docs/postgrespro/9.5/functions-string>
4. Тестер регулярных выражений - <https://www.regextester.com>
5. Интерактивный учебник по SQL - <http://www.sql-tutorial.ru/ru/content.html>
6. <https://www.sqlitetutorial.net/>
7. <https://docs.python.org/3/library/sqlite3.html> - документация по модулю sqlite3
8. <https://doc.qt.io/qtforpython/index.html> – документация
9. <https://www.pythonguis.com/pyside6/> – учебник
10. <https://pytest-qt.readthedocs.io/en/latest/intro.html> – документация по pytest-qt

4.2. Описание материально-технической базы, необходимой для осуществления образовательного процесса по программе

Таблица 26

Наименование специализированных аудиторий, кабинетов, лабораторий	Вид занятий	Наименование оборудования, программного обеспечения
Система дистанционного обучения	Лекция	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю курса необходимо наличие доступа администратора курса и оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.
Система дистанционного обучения	Практическая работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер.
Система дистанционного обучения	Самостоятельная работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер.
Система дистанционного обучения	Промежуточная (итоговая) аттестация	Слушателю необходимо наличие доступа в сеть интернет, компьютер.

5. Организация образовательного процесса

В таблице 27 описаны образовательные технологии.

Таблица 27

№ п/п	Вид занятия	Форма проведения занятий	Цель
1	Лекция	Самостоятельный просмотр видеолекций Обсуждение вопросов, возникших в результате просмотра видеолекций, выполнения практических заданий и изучения литературы.	актуализация и систематизация теоретических знаний по дисциплине
2	Практическая работа	Выполнение практических заданий. Прохождение практики в организации партнера в дистанционном формате.	осознание связей между теорией и практикой, повышение степени понимания материала
3	Самостоятельная работа	Самостоятельное изучение дополнительных материалов и литературы.	получение дополнительных теоретических знаний
4	Промежуточная (итоговая) аттестация	Выполнение заданий.	контроль освоения программы

РАБОЧАЯ ПРОГРАММА МОДУЛЯ 3

Создание web-сервисов на Python

1. Цель освоения Модуля 3

Целью реализации модуля «Создание web-сервисов на Python» является совершенствование компетенций слушателей в области создания и управления web-сервисами на основе Python, работы с различными инструментами сбора данных.

1.1. Совершенствуемые и/или приобретаемые компетенции

Компетенции, формируемые и совершенствуемые в результате обучения, представлены в таблицах 28 и 29.

Таблица 28

№	Компетенции в соответствии с профессиональным стандартом 06.001 Программист	Код компетенции
1	Способен написать программный код с использованием языков программирования, определения и манипулирования данными	ПК-1
2	Способен работать с системой контроля версий	ПК-2
3	Способен проверить и отладить программный код	ПК-5

Таблица 29а

№	Компетенции в соответствии с направлением подготовки 09.03.01 Информатика и вычислительная техника (ФГОС)	Код компетенции
1.	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	УК-3
2	Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8
3	Способен осваивать методики использования программных средств для решения практических задач	ОПК-9

Таблица 29б

№	Профессиональные компетенции (ПК) в соответствии с направлением подготовки 09.03.01 Информатика и вычислительная техника (Собственный стандарт МФТИ)	Код компетенции
1.	Способен формализовать и алгоритмизировать поставленную задачу, написать программный код с использованием языков программирования, оформить код в соответствии с установленными требованиями	ПК-3

Таблица 29в

Наименование сферы	Код и наименование профессиональной компетенции	БАЗОВЫЙ УРОВЕНЬ РАЗВИТИЯ КОМПЕТЕНЦИЙ Способность проявляется под внешним контролем / при внешней постановке задачи/ обучающийся пользуется готовыми, рекомендованными продуктами
Средства программной разработки	ПК-4 Применяет языки программирования для решения профессиональных задач	Самостоятельно применяет языки программирования. Использует настраиваемые программные инструменты для автоматизации процессов в профессиональной деятельности.
Интернет-технологии	ПК-6 Разрабатывает различные веб-ориентированные решения	Участвует под контролем опытных специалистов в разработке веб-решений, с применением фреймворков и платформ

1.2. Планируемые результаты обучения

Планируемые результаты обучения представлены в таблице 30.

Таблица 30

№	Уметь - знать	Направление подготовки 09.03.01 Информатика и вычислительная техника (ФГОС) Квалификация: бакалавр Код компетенции
1	Уметь: – выполнять проектную деятельность	УК-3
2	Уметь: – пользоваться библиотекой requests	ОПК-8
3	Знать: – методы модуля BS Уметь: – с помощью библиотеки requests извлекать информацию из интернета	ОПК-9
		Профессиональный стандарт 06.001 Программист Код компетенции
4	Знать: – что такое плейсхолдеры – правила использования Плейсхолдеров Уметь:	ПК-1

	<ul style="list-style-type: none"> – составлять список книг в каталоге при помощи URL 	
5	<p>Знать:</p> <ul style="list-style-type: none"> – основные протоколы транспортного уровня – как собирать данные из интернета – какие существуют способы для извлечения информации из интернета – что такое URL и путь – URL, view, шаблоны <p>Уметь:</p> <ul style="list-style-type: none"> – пользоваться модулем для подгрузки данных из интернета для решения собственных задач 	ПК-2
6	<p>Знать:</p> <ul style="list-style-type: none"> – как функционирует взаимодействие в интернете – что такое WEB-API – как устроена генерация ответа на HTTP-запрос <p>Уметь:</p> <ul style="list-style-type: none"> – пользоваться WEB-API 	ПК-5
		Направление подготовки 09.03.01 Информатика и вычислительная техника (Собственный стандарт МФТИ)
7	<p>Знать:</p> <ul style="list-style-type: none"> – что такое блоки <p>Уметь:</p> <ul style="list-style-type: none"> – пользоваться блоками 	ПК-3
		Матрица (модель) цифровых компетенций в части профессиональных компетенций в ИТ-сфере (Иннополис)
8	<p>Знать:</p> <ul style="list-style-type: none"> – как получить данные с помощью регулярных выражений – структуру проекта на Django <p>Уметь:</p> <ul style="list-style-type: none"> – собирать данные с помощью регулярных выражений – извлекать и изменять данные при помощи регулярных выражений и модуля BeautifulSoup; – добавлять файлы проекта на Django 	ПК-4
9	<p>Знать:</p> <ul style="list-style-type: none"> – чем отличаются два основных веб-фреймворка на Python – какие существуют web-фреймворки на Python 	ПК-6

	Уметь: <ul style="list-style-type: none"> – создавать веб-приложения на основе фреймворка Django; – настраивать взаимодействие с базами данных с помощью инструмента Django ORM; 	
--	--	--

1.3. Категория обучающихся

Программа повышения квалификации предназначена для лиц, имеющих высшее образование и желающих получить компетенции в сфере программирования на Python для своей профессиональной деятельности; студентов вузов.

1.4. Форма обучения

Очная с применением дистанционных образовательных технологий.

Программа может быть реализована в сетевой форме.

1.5. Объем программы

62 академических часа.

1.6. Режим обучения

7 недель.

2. Содержание программы

2.1. Учебный (тематический) план

Учебный (тематический) план программы представлен в таблице 31.

Таблица 31

№ п/п	Наименование разделов (модулей) и тем	Всего, час.	Лекции	Практич. работа	Самост. работа	Форма контроля
УЧЕБНЫЙ ПЛАН						
1	Общее представление о WEB	8	1,4	2	4,6	Тест
2	Сбор данных со сторонних сайтов	8	1,7	2	4,3	Тест, задание на программирование
3	Beautiful Soup и работа с API	8	2,1	2	3,9	Задание на программирование
4	Разработка WEB-приложения	32	6,4	8	17,6	Задание проекта

	Итоговая аттестация	6		4	2	зачет
	Итого:	62	11,6	18	30,4	
УЧЕБНЫЙ (ТЕМАТИЧЕСКИЙ) ПЛАН						
1	Общее представление о WEB	8	1,4	2	4,6	Тест
1.1	Основы организации компьютерных сетей и модель TCP/IP	0,5	0,5			
1.2	Транспортный уровень и его протоколы	2				
1.3	Библиотека requests	0,3	0,3			
1.4	Практика по requests	0,6	0,6			
1.5	Семинар	2		2		
1.6	Дополнительные материалы	1,1			1,1	
1.7	Тест на оценку	1,5			1,5	Тест
2	Сбор данных со сторонних сайтов	8	1,7	2	4,3	Тест, задание на программирование
2.1	Введение в обработку данных	0,2	0,2			
2.2	Поиск с помощью регулярных выражений	0,3	0,3			
2.3	Символьные классы и квантификаторы	0,5	0,5			
2.4	Сложный поиск и замена	0,7	0,7			
2.5	Семинар	2		2		
2.6	Дополнительные материалы	1			1	
2.7	Тест на оценку	1,5			1,5	Тест
2.8	Задание на программирование	1,8			1,8	Задание на программирование

3	Beautiful Soup и работа с API	8	2,1	2	3,9	Задание на программирование
3.1	Обзор методов модуля Beautiful Soup	0,2	0,2			
3.2	Сложный поиск и изменение с Beautiful Soup	0,7	0,7			
3.3	Взаимодействие с API	0,3	0,3			
3.4	Практический пример работы с API	0,9	0,9			
3.5	Семинар	2		2		
3.6	Дополнительные материалы	1,9			1,9	
3.7	Задание на программирование	2			2	Задание на программирование
4	Разработка WEB-приложения	32	6,4	8	17,6	Задание проекта
4.1	Python и WEB-фреймворки	8	1,9	2	4,1	Задание проекта
4.1.1	Обзор существующих web-фреймворков Python	0,3	0,3			
4.1.2	Архитектура web-приложения	0,2	0,2			
4.1.3	Роутинг и устройство view	0,2	0,2			
4.1.4	Установка и запуск простейшего приложения на Django	0,8	0,8			
4.1.5	Шаблонизация Django	0,4	0,4			
4.1.6	Семинар	2		2		
4.1.7	Дополнительные материалы	1,1			1,1	
4.1.8	Задание проекта	3			3	Задание проекта

4.2	Взаимодействие с базами данных с помощью Django	8	1	2	5	Задание проекта
4.2.1	Использование различных баз данных для разработки web-приложений	0,4	0,4			
4.2.2	Работа с Django ORM	0,3	0,3			
4.2.3	Реализация Django ORM в проекте	0,3	0,3			
4.2.4	Семинар	2		2		
4.2.5	Дополнительные материалы	2			2	
4.2.6	Задание проекта	3			3	Задание проекта
4.3	Основы Frontend-разработки	8	1,8	2	4,2	Задание проекта
4.3.1	Основы HTML-вёрстки	0,5	0,5			
4.3.2	Подключаем CSS	0,3	0,3			
4.3.3	JavaScript и его применение	0,2	0,2			
4.3.4	Дизайн-системы	0,3	0,3			
4.3.5	Настройка проекта с использованием дизайн-системы	0,5	0,5			
4.3.6	Семинар	2		2		
4.3.7	Дополнительные материалы	1,2			1,2	
4.3.8	Задание проекта	3			3	Задание проекта
4.4	Продвинутые возможности Django	8	1,7	2	4,3	Задание проекта
4.4.1	Автоматическое тестирование и Django	0,7	0,7			
4.4.2	Django и Telegram-боты	0,4	0,4			
4.4.3	Контейнеризация	0,3	0,3			

4.4.4	Асинхронные запросы (Celery)	0,3	0,3			
4.4.5	Семинар	2		2		
4.4.6	Дополнительные материалы	1,3			1,3	
4.4.7	Задание проекта	3			3	Задание проекта
	Итоговая аттестация	6		4	2	зачет
	Итого:	62	11,6	18	30,4	

2.2. Учебная программа

Содержание учебной программы приведено в таблице 32.

Таблица 32

№ п/п	Наименование модуля, разделов и тем	Содержание обучения, наименование и тематика практических занятий (семинаров), самостоятельной работы	Объем, ак. час.
1	Общее представление о WEB	<p>Лекция</p> <p>Основы организации компьютерных сетей и модель TCP/IP</p> <p>Транспортный уровень и его протоколы</p> <p>Библиотека requests</p> <p>Практика по requests</p> <p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Дополнительные материалы</p> <p>Тест на оценку</p>	8
2	Сбор данных со сторонних сайтов	<p>Лекция</p> <p>Введение в обработку данных</p> <p>Поиск с помощью регулярных выражений</p> <p>Символьные классы и квантификаторы</p> <p>Сложный поиск и замена</p>	8

		<p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Тест на оценку Задания на программирование (практическое задание по регулярным выражениям)</p>	
3	Beautiful Soup и работа с API	<p>Лекция</p> <p>Обзор методов модуля BeautifulSoup</p> <p>Сложный поиск и изменение с BeautifulSoup</p> <p>Работа через Web-API</p> <p>Практика работы с API</p> <p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Дополнительные материалы Задания на программирование (практическое задание по BeautifulSoup, Конвертер валют)</p>	8
4	Разработка WEB-приложения		32
4.1	Python и WEB-фреймворки	<p>Лекция</p> <p>Обзор существующих web-фреймворков Python</p> <p>Архитектура web-приложения</p> <p>Роутинг и устройство view</p> <p>Установка и запуск простейшего приложения на Django</p> <p>Шаблонизация Django</p> <p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Дополнительные материалы</p>	8

		Задание проекта	
4.2	Взаимодействие с базами данных с помощью Django	<p>Лекция</p> <p>Использование различных баз данных для разработки web-приложений</p> <p>Работа с Django ORM</p> <p>Реализация Django ORM в проекте</p> <p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Дополнительные материалы</p> <p>Задание проекта</p>	8
4.3	Основы Frontend-разработки	<p>Лекция</p> <p>Основы HTML-вёрстки</p> <p>Подключаем CSS</p> <p>JavaScript и его применение</p> <p>Дизайн-системы</p> <p>Настройка проекта с использованием дизайн-системы</p> <p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Дополнительные материалы</p> <p>Задание проекта</p>	8
4.4	Продвинутые возможности Django	<p>Лекция</p> <p>Автоматическое тестирование и Django</p> <p>Django и Telegram-боты</p> <p>Контейнеризация</p> <p>Асинхронные запросы (Celery)</p> <p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Дополнительные материалы</p>	8

		Задание проекта	
	Итоговая аттестация	зачет	6
	Итого:		62

3. Формы аттестации и оценочные материалы

3.1. Формы аттестации

Итоговая аттестация по программе является обязательной.

Форма итоговой аттестации предусматривает зачет в форме тестирования.

Оценка «зачтено» выставляется обучающемуся, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка «не зачтено» выставляется обучающемуся, 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.

Максимальная сумма, которую можно набрать, успешно выполнив все контрольные мероприятия, составляет 100 баллов. Для получения положительной оценки «зачтено» необходимо набрать не менее 30 баллов.

В зачетно-экзаменационную ведомость оценка выставляется в соответствии с нижеприведенной таблицей 33.

Таблица 33

Сумма баллов	Оценка
30-100	Зачтено
менее 30	Не зачтено

Составляющие процесса обучения, которые оцениваются в ходе обучения, и их вклад в итоговую оценку представлены в таблице 34.

Таблица 34

№ п/п	Основные показатели оценки	Вклад в итоговую оценку
1	Практические задания	30%
2	Выполнение домашних заданий	40%
3	Промежуточная (итоговая) аттестация	30%

3.2.Оценочные материалы

Таблица 35

Наименование модуля, разделов и тем	Основные показатели оценки	Формы и методы контроля и оценки	Вес задания, %
Общее представление о WEB	ОПК-8, ПК-5	Практические задания по темам лекций, тестирование	20
Сбор данных со сторонних сайтов	ОПК-9, ПК-5	Тестирование, задания на программирование (практическое задание по регулярным выражениям)	20
Beautiful Soup и работа с API	ПК-5, ПК-4	Задания на программирование (практическое задание по Beautiful Soup, Конвертер валют)	20
Разработка WEB-приложения	ПК-1, ПК-2, ПК-3, ПК-4, ПК-5, ПК-6	Задания проекта	20
Итоговая аттестация	УК-3, ОПК-8, ОПК-9, ПК-1, ПК-2, ПК-3, ПК-4, ПК-5, ПК-6	Итоговый проект	20

Примеры заданий

Тестовые задания

Вопрос 1. Что из перечисленного не может быть IP-адресом?

- а) 192.167.3.5
- б) 255.255.255.255
- в) 2001:0db8:0001:0000:0000:0ab9:C0A8:0102
- г) 192.256.2.2

Вопрос 2. Для приёма и отправки почты, как и для некоторых других действий (например, отправки HTTP-запросов), зарезервированы порты в _____ пуле.

- а) общеизвестном
- б) зарегистрированном
- в) пользовательском
- г) частном

Вопрос 3. Известно, что IPv4 адреса — это 4 блока чисел от 0 до 255 (т.е. всего доступно 256 вариантов). Сколько всего IPv4 адресов существует? Запишите количество цифр в этом числе

Вопрос 4. Даны компьютеры с двумя IP-адресами: 215.154.247.178 и 215.55.238.117. Какой должна быть маска подсети, чтобы эти компьютеры были в одной подсети?

а) /24

б) /8

в) /18

г) /215

Задание на программирование

Регулярные выражения

Данные

Для этого задания потребуется CSV-файл, скачать который можно по ссылке:

https://gist.githubusercontent.com/Grammer/4880a2e474515b6b1c5fd4b3dc1e11aa/raw/94e99ab2d2d2d8e3fc10daf50cb7af23c517a1af/enron_3000.csv

Назовите скачанный файл `enron_3000.csv`

Источник данных: [Enron Email Dataset](#)

Задание

Считайте полученный файл построчно. Из каждой строки извлеките все email-адреса, которые в ней употреблялись, и сделайте большой частотный словарь, где ключом будет email-адрес, а значением — количество употреблений этого email-адреса во всём файле.

Выведите 20 самых частотных email-адресов с их частотностью, разделённых знаком табуляции.

Обратите внимание: email-адреса расположены не только в полях адресата и адресанта, иногда они встречаются и в других полях

Пример вывода

<code>top1email@enron.com</code>	<code>100500</code>
<code>top2email@enron.com</code>	<code>100499</code>
<code>top3email@enron.com</code>	<code>100498</code>

Идеальное решение

```
import re

emails_dict = {}

regex_email = re.compile("[A-Za-z\.\d]+@[a-z\-\+]\.[a-z]+")
```

```

with open("enron_3000.csv", "r", encoding="utf-8") as f:
    for line in f.readlines():
        emails = re.findall(regex_email, line)
        for email in emails:
            emails_dict[email] = emails_dict.get(email, 0) + 1
top20_emails = sorted(emails_dict.items(), key=lambda x: x[1], reverse=True)[:21]
for top_email, top_freq in top20_emails:
    print(f"{top_email}\t{top_freq}")

```

Вывод

```

phillip.allen@enron.com 2137
pallen@enron.com      621
k..allen@enron.com   445
jsmith@austintx.com  353
stagecoachmama@hotmail.com 338
pallen70@hotmail.com 315
cbpres@austin.rr     298
mike.grigsby@enron.com 247
keith.holst@enron.com 220
ina.rangel@enron.com 178
Phillip.K.Allen@enron.com 137
matt.smith@enron.com 128
frank.ermis@enron.com 126
jacquestc@aol.com    122
gthorse@keyad.com    106
matthew.lenhart@enron.com 99
maryrichards7@hotmail.com 98
llewter@austin.rr    97
phillip.k.allen@enron.com 86
john.lavorato@enron.com 74
jay.reitmeyer@enron.com 73

```

4. Организационно-педагогические условия реализации программы

4.1. Учебно-методическое обеспечение и информационное обеспечение программы

4.1.1. Список литературы

Основная литература

1. Think Python [Электронный ресурс] – Режим доступа - <https://greenteapress.com/wp/think-python-2e/>
2. Automate the Boring Stuff with Python [Электронный ресурс] – Режим доступа - <https://automatetheboringstuff.com/>
3. Dive Into Python 3 [Электронный ресурс] – Режим доступа - <http://diveintopython3.problemsolving.io/>
4. Swaroop Chitlur. A Byte of Python [Электронный ресурс] – Режим доступа - <https://wombat.org.ua/AByteOfPython/AByteofPythonRussian-2.02.pdf> – 2020.
5. Федоров Д. Основы программирования на примере языка Python [Текст] : учебное пособие / Д. Федоров. - 2018.
6. Свейгарт Э. Автоматизация рутинных задач с помощью Python [Текст] : практическое руководство для начинающих / Эл Свейгарт. – Диалектика, 2021.

Дополнительная литература

1. Бэрри П. Изучаем программирование на Python [Текст] / П. Бэрри. – 2017.

4.1.2. Ресурсы информационно-телекоммуникационной сети "Интернет"

7. Что такое TCP/IP и зачем они нужны: <https://thecode.media/tcp-ip/>
8. TCP и UDP — в чем разница: <https://wiki.merionet.ru/seti/23/tcp-i-udp-v-chem-raznica/>
9. User-Agent: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent>
10. Тестер регулярных выражений: <https://regex101.com/>
11. HTTP Request & Response Service: <http://httpbin.org>
12. <https://docs.python-requests.org/en/latest/index.html>
13. <https://www.useragentstring.com/>
14. <https://docs.python.org/3/library/re.html>
15. <https://www.crummy.com/software/BeautifulSoup/bs4/doc.ru/bs4ru.html>
16. <https://world.openfoodfacts.org/>
17. <https://publicapis.io/>
18. <https://www.djangoproject.com/>
19. <https://awesomedjango.org/>

4.2. Описание материально-технической базы, необходимой для осуществления образовательного процесса по программе

Таблица 36

Наименование специализированных аудиторий, кабинетов, лабораторий	Вид занятий	Наименование оборудования, программного обеспечения
Система дистанционного обучения	Лекция	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю курса необходимо наличие доступа администратора курса и оборудование для проведения

		дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.
Система дистанционного обучения	Практическая работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер.
Система дистанционного обучения	Самостоятельная работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер.
Система дистанционного обучения	Промежуточная (итоговая) аттестация	Слушателю необходимо наличие доступа в сеть интернет, компьютер.

5. Организация образовательного процесса

В таблице 37 описаны образовательные технологии.

Таблица 37

№ п/п	Вид занятия	Форма проведения занятий	Цель
1	Лекция	Самостоятельный просмотр видеолекций Обсуждение вопросов, возникших в результате просмотра видеолекций, выполнения практических заданий и изучения литературы.	актуализация и систематизация теоретических знаний по дисциплине
2	Практическая работа	Выполнение практических заданий. Прохождение практики в организации партнера в дистанционном формате.	осознание связей между теорией и практикой, повышение степени понимания материала
3	Самостоятельная работа	Самостоятельное изучение дополнительных материалов и литературы.	получение дополнительных теоретических знаний
4	Промежуточная (итоговая) аттестация	Выполнение заданий.	контроль освоения программы

РАБОЧАЯ ПРОГРАММА МОДУЛЯ 4

Анализ данных в Python

1. Цель освоения Модуля 4

Целью реализации модуля «Анализ данных в Python» является совершенствование компетенций слушателей в области использования библиотек NumPy, SciPy, Pandas для работы с данными, визуализации данных, применения методов обучения с учителем и без учителя, создания нейронных сетей.

1.1. Совершенствуемые и/или приобретаемые компетенции

Компетенции, формируемые и совершенствуемые в результате обучения, представлены в таблицах 38 и 39.

Таблица 38

№	Компетенции в соответствии с профессиональным стандартом 06.001 Программист	Код компетенции
1	Способен написать программный код с использованием языков программирования, определения и манипулирования данными	ПК-1
2	Способен работать с системой контроля версий	ПК-2
3	Способен проверить и отладить программный код	ПК-5

Таблица 39а

№	Компетенции в соответствии с направлением подготовки 09.03.01 Информатика и вычислительная техника (ФГОС)	Код компетенции
1.	Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8
2	Способен осваивать методики использования программных средств для решения практических задач	ОПК-9
3	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	УК-3

Таблица 39б

№	Профессиональные компетенции (ПК) в соответствии с направлением подготовки 09.03.01 Информатика и вычислительная техника (Собственный стандарт МФТИ)	Код компетенции
1.	Способен формализовать и алгоритмизировать поставленную задачу, написать программный код с использованием языков программирования, оформить код в соответствии с установленными требованиями	ПК-3

Профессиональные компетенции (ПК) в соответствии с Матрицей (моделью) цифровых компетенций в части профессиональных компетенций в ИТ-сфере (Иннополис)

Таблица 39в

Наименование сферы	Код и наименование профессиональной компетенции	Базовый уровень развития компетенций. Компетенция проявляется в незначительной степени, по заданным шаблонам и с посторонней помощью
Средства программной разработки	ПК-4 Применяет языки программирования для решения профессиональных задач	Самостоятельно применяет языки программирования. Использует настраиваемые программные инструменты для автоматизации процессов в профессиональной деятельности.

1.2. Планируемые результаты обучения

Планируемые результаты обучения представлены в таблице 40.

Таблица 40

№	Уметь - знать	Направление подготовки 09.03.01 Информатика и вычислительная техника (ФГОС)
		Квалификация: бакалавр
		Код компетенции
1	<p>Знать:</p> <ul style="list-style-type: none"> – номинальные, порядковые, интервальные шкалы и шкалу отношений – отличия разных типов шкал. – категориальные и количественные переменные – меры распределения переменных – что такое частота распределения – что такое нормальное распределение и как его интерпретировать – свойства нормального распределения. – правило трех сигм. – дискретное, непрерывное и равномерное распределение <p>Уметь:</p> <ul style="list-style-type: none"> – загружать данные с расширениями csv., raw., xls.,xlsx., mat., xml., json – определить какие переменные принадлежат к какому типу шкалы – отличить моду, среднее, медиану – найти / вычислить моду, среднее, медиану 	ОПК-8

	<ul style="list-style-type: none"> – строить функции и интерпретировать полученный график – решать задачи по построению разных видов диаграмм (точечная диаграмма, столбиковая диаграмма, диаграмма соотношения), различать нормальное распределение, определять и удалять выбросы 	
2	<p>Знать:</p> <ul style="list-style-type: none"> – инструменты для обработки и анализа данных – меры разброса – что такое асимметрия и эксцесс – как посчитать точечную оценку и интерпретировать ее – как рассчитать доверительный интервал и интерпретировать его – центральную предельную теорему. <p>Уметь:</p> <ul style="list-style-type: none"> – вычислить дисперсию, стандартное отклонение – интерпретировать результаты вычислений – интерпретировать графики – вычислять процентиля и квартили – решать задачи на комбинаторику, сложение и произведение вероятностей, с зависимыми событиями, случайными событиями, объяснять формулу Байеса, формулу Бернулли, решать задачи на условную вероятность, задачи на математическое ожидание, дисперсию и стандартное отклонение случайной величины, воспроизводить статистическое определение вероятности 	ОПК-9
	<p>Уметь:</p> <ul style="list-style-type: none"> – построить график для демонстрации 	УК-3
		<p>Профессиональный стандарт 06.001 Программист Код компетенции</p>
3	<p>Уметь:</p> <ul style="list-style-type: none"> – определять и заменять пропущенные значения, значения типа NULL, – создавать вектор – выполнять операции сложения и вычитания векторов – выполнять операции умножения и деления векторов 	ПК-1

	<ul style="list-style-type: none"> – выполнять операцию векторное точечное произведение – вычислять норму векторов – вычислять расстояние между векторами – вычислять скалярное произведение и угол между векторами 	
4	<p>Знать:</p> <ul style="list-style-type: none"> – отличия разных расширений данных – выборку и генеральную совокупность <p>Уметь:</p> <ul style="list-style-type: none"> – выявить данные, которым можно доверять – выгружать данные – определять правильную (достаточную) выборку – интерпретировать результаты выборки и соотносить их с результатами генеральной совокупности 	ПК-2
5	<p>Знать:</p> <ul style="list-style-type: none"> – существующие источники данных <p>Уметь:</p> <ul style="list-style-type: none"> – тестировать значимость нулевой гипотезы. 	ПК-5
		Направление подготовки 09.03.01 Информатика и вычислительная техника (Собственный стандарт МФТИ)
6	<p>Уметь:</p> <ul style="list-style-type: none"> – создавать массив – вызывать элемент с помощью индекса – добавлять элемент в массив – добавлять новые строки и столбцы – удалять элемент из массива – менять элементы местами внутри массива – изменять размеры массива – транспонировать элементы массива 	ПК-3
		Матрица (модель) цифровых компетенций в части профессиональных компетенций в ИТ-сфере (Иннополис)
7	<p>Уметь:</p> <ul style="list-style-type: none"> – использовать функции <code>df.head()</code>, <code>df.info()</code>, <code>set_axis()</code>, <code>dropna()</code>, <code>isnull()</code>, <code>fillna()</code>, <code>duplicated()</code>, <code>df.dtypes()</code>, <code>to_numeric()</code>, <code>astype()</code> – создавать датафрейм и выводить его на экран. – выводить количество строк и столбцов датафрейма. 	ПК-4

	<ul style="list-style-type: none"> – выводить количество элементов в датафрейме – проводить простейший статистический анализ датафрейма с помощью describe() – выбирать один элемент из датафрейма – присваивать и заменять значения элементу датафрейма – проводить операции между различными датафреймами (переносы) – пользоваться функциями read_csv(), data.pivot_table(), count(), sum(), groupby(), merge(), sort(), reset_index(), fillna() 	
--	---	--

1.3. Категория обучающихся

Программа повышения квалификации предназначена для лиц, имеющих высшее образование и желающих получить компетенции в сфере программирования на Python для своей профессиональной деятельности; студентов вузов.

1.4. Форма обучения

Очная с применением дистанционных образовательных технологий.
Программа может быть реализована в сетевой форме.

1.5. Объем программы

37 академических часов.

1.6. Режим обучения

6 недель.

2. Содержание программы

2.1. Учебный (тематический) план

Учебный (тематический) план программы представлен в таблице 41.

Таблица 41

№ п/п	Наименование разделов (модулей) и тем	Всего, час.	Лекции	Практич. работа	Самост. работа	Форма контроля
УЧЕБНЫЙ ПЛАН						
1	Введение в анализ данных	8	1,9	2	4,1	Тесты, задания на программирование

2	Исследование данных с Python	16	3,1	4	8,9	Тесты, задания на программирование
3	Проект	7	0,5	4	2,5	Проект
	Итоговая аттестация	6		4	2	зачет
	Итого:	37	5,5	14	15,5	

УЧЕБНЫЙ (ТЕМАТИЧЕСКИЙ) ПЛАН

1	Введение в анализ данных	8	1,9	2	4,1	Тесты, задания на программирование
1.1	Виды данных	0,4	0,4			
1.2	Предобработка данных	0,4	0,4			
1.3	Инструменты работы с данными	0,5			0,5	
1.4	Тест на самопроверку по уроку	0,2			0,2	Тест
1.5	Задания на программирование	0,4			0,4	Задание на программирование
1.6	Массивы	0,4	0,4			
1.7	Векторы	0,2	0,2			
1.8	Семинар	2		2		
1.9	Дополнительные материалы	0,2			0,2	
1.10	Тест на самопроверку по уроку	0,2			0,2	Тест
1.11	Задания на программирование	0,5			0,5	Задание на программирование

1.12	Тест на оценку	0,5			0,5	Тест
1.13	Задания на оценку	1			1	Задание
1.14	Библиотека Pandas	0,5	0,5			
1.15	Тест на самопроверку по уроку	0,2			0,2	Тест
1.16	Задания на программирование	0,4			0,4	Задание на программирование
2	Исследование данных с Python	16	3,1	4	8,9	Тесты, задания на программирование
2.1	Описательные статистики. Статистика вывода	8	1,9	2	4,1	Тест, задание на программирование
2.1.1	Описательные статистики	0,4	0,4			
2.1.2	Тест на самопроверку по уроку	0,5			0,5	
2.1.3	Задания на программирование	1			1	
2.1.4	Выборка и генеральная совокупность	0,4	0,4			
2.1.5	Распределения	0,3	0,3			
2.1.6	Оценки	0,3	0,3			
2.1.7	Тестирование гипотез	0,5	0,5			
2.1.8	Семинар	2		2		

2.1. 9	Дополнительные материалы	1			1	
2.1. 10	Тест на самопроверку по уроку	0,6			0,6	Тест
2.1. 11	Задания на программирование	1			1	Задание на программирование
2.2	Методы визуализации	8	1,2	2	4,8	Тест, задание на программирование
2.2. 1	Методы визуализации	0,4	0,4			
2.2. 2	Библиотека Matplotlib	0,5	0,5			
2.2. 3	Библиотека Seaborn	0,3	0,3			
2.2. 4	Семинар	2		2		
2.2. 5	Дополнительные материалы	0,5			0,5	
2.2. 6	Тест на самопроверку	0,5			0,5	Тест
2.2. 7	Задания на программирование	0,8			0,8	Задание на программирование
2.2. 8	Тест на оценку	1			1	Тест
2.2. 9	Задания на оценку	2			2	Задание
3	Проект “Анализ данных в Python“	7	0,5	4	2,5	Проект
3.1	Задание проекта	1,5			1,5	Проект
3.2	Разбор примера проекта	0,5	0,5			

3.3	Семинар	4		4		
3.4	Дополнительные материалы	1			1	
	Итоговая аттестация	6		4	2	зачет
	Итого:	37	5,5	14	15,5	

2.2. Учебная программа

Содержание учебной программы приведено в таблице 42.

Таблица 42

№ п/п	Наименование модуля, разделов и тем	Содержание обучения, наименование и тематика практических занятий (семинаров), самостоятельной работы	Объем, ак. час.
1	Введение в анализ данных	<p>Лекция</p> <p>Виды данных Предобработка данных</p> <p>Самостоятельная работа</p> <p>Инструменты работы с данными Тест на самопроверку по уроку Задания на программирование</p> <p>Лекция</p> <p>Массивы Векторы</p> <p>Практическая работа</p> <p>Выполнение заданий по теме лекции</p> <p>Самостоятельная работа</p> <p>Дополнительные материалы Тест на самопроверку по уроку Задания на программирование Тест на оценку Задания на оценку</p> <p>Лекция</p> <p>Библиотека Pandas</p> <p>Самостоятельная работа</p> <p>Тест на самопроверку по уроку</p>	8

		Задания на программирование	
2	Исследование данных с Python		16
2.1	Описательные статистики. Статистика вывода	<p><i>Лекция</i></p> <p>Описательные статистики</p> <p><i>Самостоятельная работа</i></p> <p>Тест на самопроверку по уроку Задания на программирование</p> <p><i>Лекция</i></p> <p>Выборка и генеральная совокупность Распределения Оценки Тестирование гипотез</p> <p><i>Практическая работа</i></p> <p>Выполнение заданий по теме лекции</p> <p><i>Самостоятельная работа</i></p> <p>Дополнительные материалы Тест на самопроверку по уроку Задания на программирование</p>	8
2.2	Методы визуализации	<p><i>Лекция</i></p> <p>Методы визуализации Библиотека Matplotlib Библиотека Seaborn</p> <p><i>Практическая работа</i></p> <p>Выполнение заданий по теме лекции</p> <p><i>Самостоятельная работа</i></p> <p>Дополнительные материалы Тест на самопроверку Задания на программирование Тест на оценку Задания на оценку</p>	8
3	Проект “Анализ данных в Python“	<p><i>Самостоятельная работа</i></p> <p>Задание проекта</p> <p><i>Лекция</i></p> <p>Разбор примера проекта</p> <p><i>Практическая работа</i></p>	7

		Выполнение заданий по теме лекции <i>Самостоятельная работа</i> Дополнительные материалы	
	Итоговая аттестация	зачет	6
	Итого:		37

3. Формы аттестации и оценочные материалы

3.1. Формы аттестации

Итоговая аттестация по программе является обязательной.

Форма итоговой аттестации предусматривает зачет в форме тестирования.

Оценка «зачтено» выставляется обучающемуся, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка «не зачтено» выставляется обучающемуся, 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.

Максимальная сумма, которую можно набрать, успешно выполнив все контрольные мероприятия, составляет 100 баллов. Для получения положительной оценки «зачтено» необходимо набрать не менее 30 баллов.

В зачетно-экзаменационную ведомость оценка выставляется в соответствии с нижеприведенной таблицей 43.

Таблица 43

Сумма баллов	Оценка
30-100	Зачтено
менее 30	Не зачтено

Составляющие процесса обучения, которые оцениваются в ходе обучения, и их вклад в итоговую оценку представлены в таблице 44.

Таблица 44

№ п/п	Основные показатели оценки	Вклад в итоговую оценку
1	Практические задания	30%
2	Выполнение домашних заданий	40%
3	Промежуточная (итоговая) аттестация	30%

3.2.Оценочные материалы

Таблица 45

Наименование модуля, разделов и тем	Основные показатели оценки	Формы и методы контроля и оценки	Вес задания, %
Введение в анализ данных	ОПК-8, ОПК-9, ПК-1, ПК-2, ПК-3, ПК-4	Тесты, задания на программирование	25
Исследование данных с Python	ОПК-8, ОПК-9, ПК-5, ПК-2, ПК-3, ПК-4	Тесты, задания на программирование	25
Проект	УК-3, ОПК-8, ОПК-9, ПК-1, ПК-5	Проект	25
Итоговая аттестация	УК-3, ОПК-8, ОПК-9	Итоговый проект	25

Примеры заданий

Тестовые задания

- Какой из нижеприведенных источников данных хорошо подходит для тренировки и отработки навыка анализ данных:
 - GitHub
 - Kaggle
 - Всемирный Банк
 - Росстат
- Какое из приведенных расширений данных основано на другом языке программирования:
 - .sav
 - .json
 - .csv
 - .tsv
- Какая информация не содержится в методе `info()`:
 - размер датасета
 - имена колонок
 - количество пропущенных значений
 - все это содержится в выдаче метода
- Что делает данный код: `df.dropna(how = 'all', inplace = True)`:
 - удаляет все строки, где есть все пропущенные значение по колонкам и не перезаписывает исходный датафрейм
 - удаляет все строки, где есть хотя бы одно пропущенное значение по колонкам и перезаписывает исходный датафрейм
 - удаляет все строки, где есть хотя бы одно пропущенное значение по колонкам и не перезаписывает исходный датафрейм
 - все выше перечисленное неверно

Задание на программирование

Вам дан датафрейм:

```
import numpy as np
import pandas as pd

df = pd.DataFrame(np.nan, index=[0, 1, 2, 3], columns=['A', 'B'])
df.loc[0,"A"] = 1
df.loc[1,"A"] = 2
df.loc[2,"A"] = 3
df.loc[3,"A"] = 4
df.loc[0,"B"] = 5
df.loc[1,"B"] = 6
df.loc[2,"B"] = 7
df
```

```
   A  B
0  1.0  5.0
1  2.0  6.0
2  3.0  7.0
3  4.0  NaN
```

Выполните задания.

Задание 1

Переименуйте названия строк в последовательность чисел от 1 до 4, используя соответствующий метод pandas.

Стандартный вывод:
Нет входных данных

Ожидаемый результат:
Датафрейм с переименованными строками

```
df.set_axis([1, 2, 3, 4], axis='index')
```

```
   A  B
1  1.0  5.0
2  2.0  6.0
3  3.0  7.0
4  4.0  NaN
```

```
df.set_axis(list(range(1,5)), axis='index')
```

```
   A  B
1  1.0  5.0
2  2.0  6.0
3  3.0  7.0
4  4.0  NaN
```

Контрольный пример

`df.index[3] == 3`

Стандартный ввод

Ожидаемый результат

True

Задание 2

Определите какой тип данных у второй колонки датафрейма и выведите его на экран.

Стандартный вывод:

Нет входных данных

Ожидаемый результат:

Тип данных

Примечание: здесь есть единственно верный ответ -- float64

```
print(df.B.dtype)
```

float64

Контрольный пример

`df.B.dtype == 'float64'`

Стандартный ввод

Ожидаемый результат

True

Задание 3

Замените пропущенное значение в датафрейме числом 6 и пересохраните датафрейм.

Стандартный вывод:

Нет входных данных

Ожидаемый результат:

Датафрейм с заполненными пропусками

```
df = df.fillna(6)
```

Контрольный пример

`df.iloc[3,1] == 6`

Стандартный ввод

Ожидаемый результат

True

4. Организационно-педагогические условия реализации программы

4.1. Учебно-методическое обеспечение и информационное обеспечение программы

4.1.1. Список литературы

Основная литература

1. Think Python [Электронный ресурс] – Режим доступа - <https://greenteapress.com/wp/think-python-2e/>
2. Automate the Boring Stuff with Python [Электронный ресурс] – Режим доступа - <https://automatetheboringstuff.com/>
3. Dive Into Python 3 [Электронный ресурс] – Режим доступа - <http://diveintopython3.problemsolving.io/>
4. Problem Solving with Algorithms and Data Structures using Python [Электронный ресурс] – Режим доступа - <https://runestone.academy/runestone/static/pythonds/index.html>
5. Swaroop Chitlur. A Byte of Python [Электронный ресурс] – Режим доступа - <https://wombat.org.ua/AByteOfPython/AByteofPythonRussian-2.02.pdf> – 2020.
6. Федоров Д. Основы программирования на примере языка Python [Текст] : учебное пособие / Д. Федоров. - 2018.
7. Свейгарт Э. Автоматизация рутинных задач с помощью Python [Текст] : практическое руководство для начинающих / Эл Свейгарт. – Диалектика, 2021.
8. Лутц, Марк. Изучаем Python: авторитетный курс объектно-ориентированного программирования / Марк Лутц; перевод с английского Ю. Н. Артеменко. – 5-е изд. – Москва: Диалектика; Санкт-Петербург: Диалектика, 2020.
9. Reitz, K., Schlusser, T. The Hitchhiker's Guide to Python: Best Practices for Development. – O'Reilly, 2016. – URL: docs.python-guide.org

Дополнительная литература

1. Маккинни У. Python и анализ данных [Текст] / У. Маккинни. – 2015.
2. Бэрри П. Изучаем программирование на Python [Текст] / П. Бэрри. – 2017.
3. Савельев В. Статистика и коттики [Текст] / Владимир Савельев. – 2018.
4. Бослав С. Статистика для всех [Текст] / Сара Бослав. – 2015.
5. Хамидуллин Р. Я. Теория вероятностей и математическая статистика [Текст] / Р.Я. Хамидуллин. – 2020.

4.1.2. Ресурсы информационно-телекоммуникационной сети "Интернет"

1. Введение в анализ данных с помощью Pandas - <https://habr.com/ru/post/196980/>
2. Открытые датасеты - <https://www.kaggle.com/datasets>
3. Федеральная служба государственной статистики - <https://rosstat.gov.ru/>
4. Игровой тренажер по Python: py.checkio.org
5. NumPy documentation — NumPy v1.24 Manual <https://numpy.org/doc/stable/>
6. MP3 to Numpy and back | Kaggle <https://www.kaggle.com/code/gabrielmilan/mp3-to-numpy-and-back>
7. [User Guide — pandas 2.0.2 documentation \(pydata.org\)](https://pandas.pydata.org/docs/user_guide.html)
8. [Pandas Pivot Table Explained - Practical Business Python \(pbpython.com\)](https://pbpython.com/pandas-pivot-table-explained/)
9. [Моем датасет: руководство по очистке данных в Python \(proglib.io\)](https://proglib.io/ru/2019/05/01/moem-dataset-rukovodstvo-po-ochistke-dannykh-v-python/)
10. [pandas documentation — pandas 1.5.3 documentation](https://pandas.pydata.org/docs/1.5.3.html)
11. [Наглядная шпаргалка по операциям с DataFrame в pandas для data wrangling и не только \(tproger.ru\)](https://tproger.ru/ru/2019/05/01/naqliadnaya-shpargalka-po-operatsiyam-s-dataframe-v-pandas-dlya-data-wrangling-i-ne-toilko/)
12. [Statistical functions \(scipy.stats\) — SciPy v1.10.1 Manual](https://docs.scipy.org/doc/scipy-1.10.1/)

13. [Understanding Confidence Intervals | Easy Examples & Formulas \(scribbr.com\)](#)
14. [6 Steps to Evaluate a Statistical Hypothesis Testing \(enago.com\)](#)
15. [17 Statistical Hypothesis Tests in Python \(Cheat Sheet\) - MachineLearningMastery.com](#)
16. [Matplotlib documentation](#) — [Matplotlib 3.7.1 documentation](#)
17. [Animation — Matplotlib 3.7.1 documentation](#)
18. [Example gallery — seaborn 0.12.2 documentation](#)

4.2. Описание материально-технической базы, необходимой для осуществления образовательного процесса по программе

Таблица 46

Наименование специализированных аудиторий, кабинетов, лабораторий	Вид занятий	Наименование оборудования, программного обеспечения
Система дистанционного обучения	Лекция	Слушателю необходимо наличие доступа в сеть интернет, компьютер. Преподавателю курса необходимо наличие доступа администратора курса и оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.
Система дистанционного обучения	Практическая работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер.
Система дистанционного обучения	Самостоятельная работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер.
Система дистанционного обучения	Промежуточная (итоговая) аттестация	Слушателю необходимо наличие доступа в сеть интернет, компьютер.

5. Организация образовательного процесса

В таблице 47 описаны образовательные технологии.

Таблица 47

№ п/п	Вид занятия	Форма проведения занятий	Цель
1	Лекция	Самостоятельный просмотр видеолекций Обсуждение вопросов, возникших в результате просмотра видеолекций, выполнения практических заданий и изучения литературы.	актуализация и систематизация теоретических знаний по дисциплине
2	Практическая работа	Выполнение практических заданий.	осознание связей между теорией и практикой, повышение степени понимания материала

		Прохождение практики в организации партнера в дистанционном формате.	
3	Самостоятельная работа	Самостоятельное изучение дополнительных материалов и литературы.	получение дополнительных теоретических знаний
4	Промежуточная (итоговая) аттестация	Выполнение заданий.	контроль освоения программы

РАЗДЕЛ 3. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ «Программирование на Python»

Оценка качества освоения программы включает текущую, промежуточную и итоговую аттестацию обучающихся.

Текущая аттестация служит для оценки объема и уровня усвоения слушателем учебного материала одного модуля программы и применяется в рамках текущего контроля успеваемости слушателя. Текущая аттестация включает прохождение студентом входного, промежуточного и итогового ассесментов Университета Иннополис.

Промежуточная аттестация слушателей включает сдачу зачетов по модулям, предусмотренным учебно-тематическим планом.

Итоговая аттестация обучающихся проходит в формате демонстрационного экзамена с участием представителей профильных организаций работодателей. Итоговая аттестация может проходить в форме защиты проекта, в ходе которого обучающиеся используют и демонстрируют цифровые компетенции, предусмотренные в процессе освоения ДПП ПП.

Контроль качества знаний осуществляется преподавателем.

Текущий контроль успеваемости слушателей проводится на платформе дистанционного обучения и на платформе Университета Иннополис.

Объектом оценивания выступают: формируемые знания и умения.

Итоговая аттестация слушателей проводится в сроки, предусмотренные учебным планом и графиком учебного процесса.

Форма и условия аттестационных испытаний определяются настоящей программой.

Защита итогового проекта проводится слушателями после получения допуска к итоговой аттестации. Доступ к прохождению итоговой аттестации предоставляется слушателям, успешно прошедшим промежуточную аттестацию по всем модулям программы.

Защита проекта сопровождается презентацией и докладом об основных этапах реализации проекта, а также демонстрацией прототипа проектного решения. Допускается защита проекта, как в группах, так и индивидуально. Проектное решение должно отвечать критериям актуальности, законченности, а также возможности интеграции его компонентов в иные системы и сервисы. Возможно проведение очной (онлайн в формате видеоконференции) или офлайн (видеозапись) защиты проекта.

Результаты итоговых аттестационных испытаний определяются отметками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно» и объявляются после оформления в установленном порядке протоколов заседаний итоговой аттестационной комиссии.

Оценка «отлично (5)» выставляется обучающемуся, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;

оценка «хорошо (4)» выставляется обучающемуся, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает неточности в ответе или делает несущественные ошибки при решении задач;

оценка «удовлетворительно (3)» выставляется обучающемуся, показавшему фрагментарный, разрозненный характер знаний, неточные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, если при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

оценка «неудовлетворительно (2)» выставляется обучающемуся, 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.

В случае несогласия с результатами итоговой аттестации слушатель имеет право подать апелляцию в аттестационную комиссию на имя председателя.

Апелляция принимается в день сдачи итогового экзамена сразу после объявления оценок.

В ходе рассмотрения апелляции проверяется правильность и аргументированность выставленной оценки. При необходимости ИАК принимает решение о возможности пересдачи итогового экзамена или повторной защиты выпускной квалификационной работы. Итоги рассмотрения апелляции вносятся в соответствующий протокол в раздел «Особое мнение комиссии».

ИАК рассматривает апелляцию по итоговым экзаменам и сообщает свое решение не позднее, чем через сутки с момента подачи апелляции. При принятии решения о возможности пересдачи итогового экзамена комиссией анализируется лист ответа слушателя и при необходимости ведется беседа по дополнительным вопросам. Решение ИАК по апелляции является окончательным.

Составляющие процесса обучения, которые оцениваются в ходе обучения, и их вклад в итоговую оценку представлены в таблице 48 ниже.

Таблица 48

№ п/п	Основные показатели оценки	Вклад в итоговую оценку, %
1	Промежуточная аттестация по модулю 1	20
2	Промежуточная аттестация по модулю 2	20
3	Промежуточная аттестация по модулю 3	20
4	Промежуточная аттестация по модулю 4	20
5	Итоговая аттестация	20

Самостоятельная работа слушателей подразделяется на аудиторную и внеаудиторную. Аудиторную самостоятельную работу составляют практические задания, которые выполняются слушателями во время учебных занятий, результаты ее выполнения проверяются и оцениваются преподавателем в учебном процессе.

Внеаудиторная самостоятельная работа включает формы: изучение дополнительной литературы, подготовка итоговых проектов по модулям, подготовка проекта к итоговой аттестации.

Основными критериями качества организации самостоятельной работы служит наличие контроля результатов самостоятельной работы.

Основными современными формами организации самостоятельной работы слушателей являются творческие работы и работа с информационными компьютерными технологиями.

РАЗДЕЛ 4. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

«Программирование на Python»

4.1. Материально-технические условия и организационное обеспечение реализации дополнительной профессиональной программы профессиональной переподготовки

Для эффективного осуществления образовательного процесса необходимы

Таблица 49

Наименование специализированных аудиторий, кабинетов, лабораторий	Вид занятий	Наименование оборудования, программного обеспечения
Система дистанционного обучения	Лекция	<p>Слушателю необходимо наличие доступа в сеть интернет, компьютер.</p> <p>Преподавателю курса необходимо наличие доступа администратора курса и оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.</p>
Система дистанционного обучения	Практическая работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер.
Система дистанционного обучения	Самостоятельная работа	Слушателю необходимо наличие доступа в сеть интернет, компьютер.
Система дистанционного обучения	Промежуточная (итоговая) аттестация	<p>Слушателю необходимо наличие доступа в сеть интернет, компьютер.</p> <p>Членам аттестационной комиссии необходимо оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.</p>

В таблице 50 описаны образовательные технологии.

Таблица 50

№ п/п	Вид занятия	Форма проведения занятий	Цель
1	Лекция	Самостоятельный просмотр видеолекций Обсуждение вопросов, возникших в результате просмотра видеолекций, выполнения практических заданий и изучения литературы.	актуализация и систематизация теоретических знаний по дисциплине
2	Практическая работа	Выполнение практических заданий. Прохождение практики в организации партнера в дистанционном формате.	осознание связей между теорией и практикой, повышение степени понимания материала
3	Самостоятельная работа	Самостоятельное изучение дополнительных материалов и литературы.	получение дополнительных теоретических знаний
4	Промежуточная (итоговая) аттестация	Выполнение заданий. Разработка и защита итогового проекта	контроль освоения программы

4.2. Учебно-методическое обеспечение реализации дополнительной профессиональной программы профессиональной переподготовки

4.2.1. Учебно-методическое обеспечение и информационное обеспечение программы

Основная литература

1. Think Python [Электронный ресурс] – Режим доступа - <https://greenteapress.com/wp/think-python-2e/>
2. Automate the Boring Stuff with Python [Электронный ресурс] – Режим доступа - <https://automatetheboringstuff.com/>
3. Dive Into Python 3 [Электронный ресурс] – Режим доступа - <http://diveintopython3.problemsolving.io/>
4. Problem Solving with Algorithms and Data Structures using Python [Электронный ресурс] – Режим доступа - <https://runestone.academy/runestone/static/pythonds/index.html>
5. Swaroop Chitlur. A Byte of Python [Электронный ресурс] – Режим доступа - <https://wombat.org.ua/AByteOfPython/AByteofPythonRussian-2.02.pdf> – 2020.

6. Федоров Д. Основы программирования на примере языка Python [Текст] : учебное пособие / Д. Федоров. - 2018.
7. Свейгарт Э. Автоматизация рутинных задач с помощью Python [Текст] : практическое руководство для начинающих / Эл Свейгарт. – Диалектика, 2021.
8. Роберт Мартин. Чистый код: создание, анализ и рефакторинг.
9. Марк Саммерфилд. Python на практике.
10. Гарри Персиваль, Боб Грегори. Паттерны разработки на Python.
11. Прохоренок Н.А., Дронов В.А. Python 3 и PyQt6. Разработка приложений. – СПб.: БХВ-Петербург, 2023
12. Harry J.W. Percival. "Test-Driven Web Development with Python": <http://www.obeythetestinggoat.com/pages/book.html#toc>
13. Книга издательства O'Reilly (доступна онлайн, англ.): <https://docs.python-guide.org/>
14. Лутц, Марк. Изучаем Python: авторитетный курс объектно-ориентированного программирования / Марк Лутц; перевод с английского Ю. Н. Артеменко. – 5-е изд. – Москва: Диалектика; Санкт-Петербург: Диалектика, 2020.
15. Учебная статья: <https://docs.python.org/3/tutorial/errors.html>
16. Reitz, K., Schlusser, T. The Hitchhiker's Guide to Python: Best Practices for Development. – O'Reilly, 2016. – URL: docs.python-guide.org

Дополнительная литература

1. Бэрри П. Изучаем программирование на Python [Текст] / П. Бэрри. – 2017.
2. Маккинни У. Python и анализ данных [Текст] / У. Маккинни. – 2015.

4.2.2. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

1. Бэрри П. Изучаем программирование на Python [Текст] / П. Бэрри. – 2017.
2. Савельев В. Статистика и котика [Текст] / Владимир Савельев. – 2018.
3. Бослав С. Статистика для всех [Текст] / Сара Бослав. – 2015.
4. Хамидуллин Р. Я. Теория вероятностей и математическая статистика [Текст] / Р.Я. Хамидуллин. – 2020.
5. Маккинни У. Python и анализ данных [Текст] / У. Маккинни. – 2015.

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

1. Официальный сайт Python - <https://www.python.org/>
2. Введение в анализ данных с помощью Pandas - <https://habr.com/ru/post/196980/>
3. Среда разработки IDE PyCharm - <https://www.jetbrains.com/pycharm/>
4. PEP 8 – Style Guide for Python Code - <https://peps.python.org/pep-0008/>
5. Что такое TCP/IP и зачем они нужны: <https://thecode.media/tcp-ip/>
6. TCP и UDP — в чем разница: <https://wiki.merionet.ru/seti/23/tcp-i-udp-v-chem-raznica/>

7. User-Agent: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent>
8. Тестер регулярных выражений: <https://regex101.com/>
9. HTTP Request & Response Service: <http://httpbin.org>
10. PySide6.QtWidgets - <https://doc.qt.io/qtforpython/PySide6/QtWidgets/index.html>
11. Введение в анализ данных с помощью Pandas - <https://habr.com/ru/post/196980/>
12. Открытые датасеты - <https://www.kaggle.com/datasets>
13. Федеральная служба государственной статистики - <https://rosstat.gov.ru/>
14. Официальная документация (англ.): <https://docs.python.org/3/>
15. Документация по исключениям: <https://docs.python.org/3/library/exceptions.html>
16. Модель GitFlow: <https://nvie.com/posts/a-successful-git-branching-model/>
17. Современный подход к версионированию для CI/CD: <https://www.atlassian.com/continuous-delivery/continuous-integration/trunk-based-development>
18. Документация Postgres про сравнение строк - <https://postgrespro.ru/docs/postgrespro/9.5/functions-matching>
19. Документация Postgres про другие функции работы со строками - <https://postgrespro.ru/docs/postgrespro/9.5/functions-string>
20. Интерактивный учебник по SQL - <http://www.sql-tutorial.ru/ru/content.html>
21. <https://www.sqlitetutorial.net/>
22. <https://docs.python.org/3/library/sqlite3.html> - документация по модулю sqlite3
23. <https://doc.qt.io/qtforpython/index.html> – документация
24. <https://www.pythonguis.com/pyside6/> – учебник
25. <https://pytest-qt.readthedocs.io/en/latest/intro.html> – документация по pytest-qt
26. <https://docs.python-requests.org/en/latest/index.html>
27. <https://www.useragentstring.com/>
28. <https://docs.python.org/3/library/re.html>
29. <https://www.crummy.com/software/BeautifulSoup/bs4/doc.ru/bs4ru.html>
30. <https://world.openfoodfacts.org/>
31. <https://publicapis.io/>
32. <https://www.djangoproject.com/>
33. <https://awesomedjango.org/>
34. Игровой тренажер по Python: py.checkio.org
35. NumPy documentation — NumPy v1.24 Manual <https://numpy.org/doc/stable/>
36. MP3 to Numpy and back | Kaggle <https://www.kaggle.com/code/gabrielmilan/mp3-to-numpy-and-back>
37. [User Guide — pandas 2.0.2 documentation \(pydata.org\)](https://pandas.pydata.org/pandas-docs/stable/user_guide.html)
38. [Pandas Pivot Table Explained - Practical Business Python \(pbpython.com\)](https://pbpython.com/pandas-pivot-table-explained/)
39. [Моем датасет: руководство по очистке данных в Python \(proglib.io\)](https://proglib.io/ru/guide/cleaning-data/)
40. [pandas documentation — pandas 1.5.3 documentation](https://pandas.pydata.org/pandas-docs/stable/10min.html)
41. [Наглядная шпаргалка по операциям с DataFrame в pandas для data wrangling и не только \(tproger.ru\)](https://tproger.ru/ru/guide/cleaning-data/)
42. [Statistical functions \(scipy.stats\) — SciPy v1.10.1 Manual](https://docs.scipy.org/doc/scipy/reference/stats.html)
43. [Understanding Confidence Intervals | Easy Examples & Formulas \(scribbr.com\)](https://www.scribbr.com/statistics/confidence-intervals/)
44. [6 Steps to Evaluate a Statistical Hypothesis Testing \(enago.com\)](https://enago.com/6-steps-to-evaluate-a-statistical-hypothesis-testing/)
45. [17 Statistical Hypothesis Tests in Python \(Cheat Sheet\) - MachineLearningMastery.com](https://machinelearningmastery.com/17-statistical-hypothesis-tests-in-python-cheat-sheet/)
46. [Matplotlib documentation — Matplotlib 3.7.1 documentation](https://matplotlib.org/3.7.1/users/first_steps.html)
47. [Animation — Matplotlib 3.7.1 documentation](https://matplotlib.org/3.7.1/users/animation.html)
48. [Example gallery — seaborn 0.12.2 documentation](https://seaborn.pydata.org/seaborn-0.12.2/tutorial.html)

5. Составители программы

Чистов Анатолий Александрович - Ассистент кафедры информатики и вычислительной математики МФТИ

Максимова Дарья Максимовна - Инженер по тестированию, Toloka.ai (ex-Яндекс Маркет, Альфа-банк)

Тарасенко Георгий Константинович - Стажер-исследователь ИнИИ НИУ ВШЭ

Машевская Екатерина Анатольевна - старший методист отдела продуктовой разработки, Центр «Пуск» МФТИ.

Зубцова Жанна Исхаковна – эксперт отдела сопровождения образовательных программ, Центр «Пуск» МФТИ.

Согласовано

Эксперт ОСОП



Ж. И. Зубцова

Согласовано

Заместитель директора (Центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск")



А. И. Рыбакова