
УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«07» марта 2024 г.
Протокол №1

**Рабочая программа дисциплины (модуля)
«Разработка на Python»**

Направление подготовки: 02.03.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Математика и компьютерные науки

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Срок освоения программы: 4 года

Год набора: 2024

**Москва
2024**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения	4
3. Тематический план	5
4. Содержание дисциплины (модуля)	5
5. Учебно-методическое обеспечение	6
6. Материально-техническое обеспечение	6
7. Методические и оценочные материалы	8

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Разработка на Python» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по специальности 02.03.01 Математика и компьютерные науки, профиль Математика и компьютерные науки, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 807 от 23.08.2017 года.

Изучение дисциплины (модуля) дает умение программирования, используемым в различных областях, таких как веб-разработка, анализ данных и машинное обучение. Освоение Python позволяет эффективно решать сложные задачи в профессиональной деятельности.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 02.03.01 Математика и компьютерные науки, профиль Математика и компьютерные науки и входит в обязательную часть Блока 1.

Дисциплина (модуль) изучается на 1 курсе в 1 семестре.

Цель изучения дисциплины (модуля): формирование у студентов навыков программирования на языке Python, включая разработку, отладку и оптимизацию приложений.

Задачи изучения дисциплины (модуля):

- освоение основ языка Python и принципы структурирования кода;
- изучение основных структур данных и методов работы с ними;
- ознакомление с фундаментальными концепциями объектно-ориентированного и функционального программирования;
- понимание базовых принципов анализа и визуализации данных;
- разрабатывание и отлаживание программы на Python;
- структурирование решений с использованием функций и классов;
- выполнение базовых обработок и анализ данных;
- создание визуализации и дашборды;
- проведение анализа данных и выявление закономерности;
- представление результатов с помощью интерактивных визуализаций и дашбордов.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
ПК-3.	Способен применять методы математического и алгоритмического моделирования для решения как теоретических, так и практических задач в рамках профессиональной деятельности	ПК-3.1.	Знает основные методы математического и алгоритмического моделирования, а также их применение для решения теоретических и прикладных задач
		ПК-3.2.	Умеет применять методы математического и алгоритмического моделирования для анализа и решения различных задач в области математики и компьютерных наук
		ПК-3.3.	Имеет опыт использования методов математического и алгоритмического моделирования при решении теоретических и прикладных задач в профессиональной деятельности
ПК-4.	Способен разрабатывать программное обеспечение для решения прикладных задач в области искусственного интеллекта под руководством более опытного специалиста	ПК-4.1.	Знает основные принципы разработки программного обеспечения и методы решения прикладных задач в сфере искусственного интеллекта
		ПК-4.2.	Умеет разрабатывать программное обеспечение под руководством специалистов более высокой категории, используя современные технологии и инструменты
		ПК-4.3.	Имеет опыт участия в разработке программного обеспечения для решения прикладных задач в команде под руководством более опытных специалистов

3. Тематический план

№п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		<i>Очная форма</i>				
		Контактная работа		Контроль	Самостоятельная работа	
Лекции	Семинары (практические занятия)					
1	Основы программирования на языке Python		30		42	Домашнее задание
2	Анализ и визуализация данных		30		42	Домашнее задание
	<i>Зачет с оценкой</i>			8		Проект
	<i>Итого:</i>		<i>60</i>	8	<i>84</i>	
	<i>Объем дисциплины (модуля) (в ак. ч.)</i>	<i>152</i>				
	<i>Объем дисциплины (модуля) (в зач. ед.)</i>	<i>4</i>				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Основы программирования на языке Python	Конструкции языка Python. Стандартные коллекции языка Python. Объявление и использование функций. Генераторы, декораторы и обработка ошибок. Основы объектно-ориентированного программирования. Контекстные менеджеры и работа с файлами
2	Анализ и визуализация данных	Массивы numpy и работа с векторными данными. Датафреймы pandas: чтение и обработка данных. Простая визуализация средствами matplotlib. Интерактивная визуализация средствами plotly

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Чернышев, С. А. Основы программирования на Python : учебник для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 349 с. — (Высшее образование). — ISBN 978-5-534-17139-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/567821>.

Дополнительная литература:

1. Гниденко, И. Г. Технологии и методы программирования : учебник для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 241 с. — (Высшее образование). — ISBN 978-5-534-18130-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/581329>.

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа к в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование»	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		

AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Разработка на Python» в рамках текущего контроля успеваемости используются такие виды учебной работы, как семинары, проекты и домашние задания, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Участие в семинаре – активная работа студента на семинаре, его ответы на вопросы преподавателя и участие в дискуссии.

Для успешного участия в семинаре студентам рекомендуется заранее ознакомиться с темой обсуждения, прочитать необходимые материалы и подготовить вопросы. Важно активно слушать и вовлекаться в дискуссию, высказывая свои мнения и аргументируя их. При ответах на вопросы преподавателя стоит быть уверенным, четким и логичным, опираясь на изученный материал. Также полезно поддерживать диалог с однокурсниками, чтобы обогатить обсуждение и расширить свои знания.

Проект – это целенаправленная деятельность, имеющая определенные цели, задачи и временные рамки, в результате которой создается уникальный продукт или услуга.

Для успешной подготовки проекта рекомендуется следовать выполнять следующие рекомендации:

- четко определите цель и задачи проекта, чтобы понимать, какой результат вы хотите достичь;
- составьте план работы, разбив проект на этапы с указанием сроков выполнения каждого из них;
- используйте разнообразные источники информации и инструменты для исследования темы, чтобы обеспечить качественную основу для вашего проекта;
- регулярно проверяйте прогресс и вносите коррективы в план, если это необходимо, чтобы оставаться на правильном пути к завершению проекта.

Домашнее задание – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал. Использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине «Разработка на Python»

Оценивание уровня учебных достижений, обучающихся по дисциплине (модулю), осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме *зачета с оценкой*, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом:

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Зачтено	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину. Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
9	Отлично	Зачтено	
8	Отлично	Зачтено	
7	Хорошо	Зачтено	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая
6	Хорошо	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
5	Удовлетворительно	Зачтено	Студент обладает базовыми знаниями по дисциплине, но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	Зачтено	
3	Не сдан	Не зачтено	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	Не зачтено	
1	Не сдан	Не зачтено	

Дисциплина (модуль) «Разработка на Python» оценивается следующим образом:

Активность	Вес	Количество	Описание
Аудиторная работа	30%	12	Решение задач на семинарах с применением автопроверки и с обратной связью от преподавателя
Домашние задания	20%	12	Набор задач по темам недели
Проекты	50%	2	Каждый проект разбивается на этапы и оценивается по заданным критериям

Формула расчёта итоговой оценки по дисциплине (модулю) «Разработка на Python»: $\langle 0,3 \times \text{среднее за аудиторную работу} + 0,20 \times \text{среднее за домашние задания} + 0,5 \times \text{среднее за проекты} \rangle$.

Для получения зачета нужно получить 4 или более баллов.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Задача 1. Создание класса продукта

Нужно создать программу для учёта товаров в отделе закупок.

- Добавь класс `Product`, который будет содержать атрибуты `name` (название товара) и `price` (цена товара).
- Добавь метод `display_info`, который будет выводить информацию о товаре.

Пример использования:

```
[ ] product = Product('Телевизор', 30000)
    product.display_info()
```

Ожидаемый вывод:

```
Название: Телевизор, Цена: 30000
```

Задача 2. Управление списком сотрудников

Нужно создать список сотрудников компании.

- Создай класс `Employee`, который содержит атрибуты `name` и `position`.
- Затем создай список из трёх сотрудников и выведи информацию о каждом из них.
- Ожидаемый вывод:
- Имя: Анна, Должность: Менеджер
Имя: Сергей, Должность: Инженер
Имя: Елена, Должность: Аналитик

Задача 3. Работа с контекстным менеджером для файлов

В файле `numbers.txt` записаны числа, каждое с новой строки. Нужно прочитать эти числа, вычислить их сумму и вывести результат.

Прочитай числа из файла `numbers.txt`, вычисли их сумму и выведи результат.

Ожидаемый вывод:

```
Сумма чисел: <сумма чисел из файла>
```

Задача 4. Чтение данных из CSV-файла

Компания хочет быстро посмотреть данные о сотрудниках, хранящиеся в CSV-файле. Нужно прочитать данные из файла и вывести информацию о каждом сотруднике.

CSV-файл `employees.csv` содержит данные о сотрудниках с колонками:

- `name` (имя),
- `age` (возраст),
- `position` (должность).

Прочитай этот файл и выведи информацию о каждом сотруднике в формате:

```
Имя: [имя], Возраст: [возраст], Должность: [должность]
```

Пример данных в файле:

```
name, age, position
Алексей, 30, Разработчик
Ирина, 25, Аналитик
Владимир, 35, Менеджер
```

Ожидаемый вывод:

```
Имя: Алексей, Возраст: 30, Должность: Разработчик
Имя: Ирина, Возраст: 25, Должность: Аналитик
Имя: Владимир, Возраст: 35, Должность: Менеджер
```

Задача 5. Сохранение объектов класса в JSON

Дмитрий — разработчик приложения для заметок. Он хочет сохранить список заметок в JSON-файл и затем загрузить их при запуске программы.

Создай класс `Note` с атрибутами:

- `title` (заголовок),
- `content` (содержимое).

Создай список заметок и сохрани их в JSON-файл `notes.json`. Затем загрузи данные из файла и выведи информацию о каждой заметке.

Ожидаемый вывод:

```
Заголовок: Заметка 1, Содержимое: Это первая заметка
Заголовок: Заметка 2, Содержимое: Это вторая заметка
```

Примерные задания по проекту

Проект: Визуализация для веб-сервиса по предсказанию неблагоприятных погодных условий для путешественников

Путешествия становятся ещё удобнее, если заранее знать, что ждёт на маршруте. Визуальное представление погодных условий по пути может сделать планирование ещё более простым и наглядным. Представь себе: ты смотришь на карту с прогнозами для каждой точки маршрута или видишь графики, которые показывают, как будет меняться погода через несколько дней. Это делает прогноз не только полезным, но и понятным для всех.

Цель. Разработать удобные визуализации и прогнозы погоды для разных точек маршрута и временных интервалов.

Основные задачи проекта

- Установить необходимые библиотеки для работы с `Plotly`, `Dash`.
- Добавить в веб-интерфейс `Dash` для визуализации и прогнозов погоды для разных точек маршрута и временных интервалов.

Проект реализуется на данных [AccuWeather](#).

Если любопытно, то подробнее о создании веб-сервисов с использованием `Flask`, `Plotly` и `Dash` можно почитать [тут](#), также полезно будет почитать форум [здесь](#), а если очень хочется, то можно посмотреть на то, как интегрировать это все дело в `Jupyter-ноутбуки` в [этом репозитории](#).

Формат сдачи проекта

Чтобы результаты работы были понятны другим (коллегам, заказчикам или даже просто тебе через месяц), придерживайся следующих правил:

- Выплоняй проект в IDE
 - Делай точечные коммиты для отслеживания изменений в проекте
 - Выложи решение на GitHub и сдай на проверку ссылку на репозиторий. Проверь что репозиторий публичный и доступен для просмотра.
 - Для ответов на вопросы в заданиях создай в проекте `.txt` или `.md` файл
-

Оценивание проекта

Проект оценивается преподавателем на основе критериев, которые указаны под каждым заданием. Баллы за выполнение всех заданий суммируются и конвертируются в итоговую оценку по **10-балльной** шкале.

Максимальное количество баллов за проект — **70 баллов**

Таблица конвертации 70-балльной оценки в 10-балльную шкалу:

Баллы	Оценка
64-70	10
57-63	9
50-56	8
43-49	7
36-42	6
29-35	5
22-28	4
15-21	3
8-14	2
1-7	1
0	0

Настройка окружения

Установи необходимые библиотеки: Plotly, Dash, Убедись, что твой сервис из проекта 2 работает корректно.

`keyboard_arrow_down`

Визуализация данных с помощью Plotly и Dash

Важной частью проекта является визуализация погодных данных для наглядного представления пользователю. Это задание подразумевает использование Plotly и Dash для создания интерактивных графиков.

Что нужно сделать:

- Используй библиотеку Plotly для построения графиков по полученным данным о погоде (например, температура, скорость ветра, вероятность осадков).
- Интегрируй графики в веб-интерфейс с использованием Dash. Dash позволяет создавать интерактивные элементы, такие как переключение временных интервалов или разные типы погодных данных.
- Сделай графики интерактивными, чтобы пользователь мог выбирать, какие параметры он хочет видеть.

Ответь на вопросы:

1. Какие графики лучше всего подходят для визуализации погодных данных? Объясни свой выбор.
2. Как можно улучшить пользовательский опыт с помощью интерактивных графиков?

Критерии оценки

1. Создание графиков с Plotly — 8 баллов:

- Построены корректные и наглядные графики для отображения данных о погоде.
- Графики позволяют пользователю взаимодействовать с данными.

2. Интеграция с Dash — 12 баллов:

- Графики корректно интегрированы в веб-интерфейс через Dash.
- Пользователь может легко переключаться между разными погодными

параметрами.

3. Ответы на вопросы — 5 баллов:

- Описаны подходящие типы графиков и обоснованы решения по их выбору.
- Разъяснены методы интерактивной работы с графиками и их интеграция.

Максимальный балл за задание — 25 баллов

Оптимизация созданного веб-интерфейса: добавление прогноза для нескольких временных интервалов и маршрута с несколькими промежуточными точками

На этом этапе мы расширим функциональность сервиса, добавив возможность отображать прогноз погоды на несколько временных интервалов (например, на 3 дня или 5 дней). Помимо этого, мы сделаем так, чтобы пользователь мог иметь возможность вводить не только начальную и конечную точку маршрута, но и промежуточные остановки.

Что нужно сделать:

- Настрой запрос к AccuWeather API для получения прогноза погоды на несколько дней вперёд.
- Собери данные для нескольких выбранных временных интервалов.
- Построй графики или таблицы, которые отображают прогноз погоды на каждый день отдельно (температура, осадки, ветер).
- Добавь возможность переключаться между временными интервалами (например, пользователь может выбрать, хочет ли он видеть прогноз на 1 день, на 3 дня или на 5 дней).
- Модифицируй HTML-форму, чтобы пользователь мог добавлять несколько промежуточных точек маршрута.
- Для каждой введённой точки маршрута отправь запрос к AccuWeather API и получи прогноз погоды.

- Обработай данные для всех точек маршрута (начальная, конечная и промежуточные).
- Построй графики или таблицы, которые отображают прогноз погоды для всех точек маршрута. Это может быть список точек с прогнозами или интерактивная карта с отметками.

Критерии оценки

- 1. Получение данных на разные интервалы — 4 балла:**
 - Данные получены для нескольких временных интервалов.
- 2. Отображение данных по дням — 6 баллов:**
 - Прогноз погоды отображён в понятной форме (графики или таблицы).
 - Пользователь может легко переключаться между интервалами прогноза.
- 3. Добавление нескольких точек маршрута — 6 баллов:**
 - Форма для ввода маршрута поддерживает несколько точек, включая промежуточные остановки.
 - Веб-интерфейс позволяет динамически добавлять промежуточные точки.
- 4. Запрос данных для всех точек — 5 баллов:**
 - Прогноз погоды корректно получен для всех точек маршрута, включая промежуточные.
 - Данные для каждой точки обработаны и переданы в модель.
- 5. Отображение результатов для всех точек — 6 баллов:**
 - Прогноз для каждой точки маршрута отображён в удобной форме (графики, таблицы или карта).

Максимальный балл, который можно получить за задание: **27 баллов**

Улучшение визуализации и дополнительная функциональность

На этом этапе мы попробуем доработать веб-сервис, добавив дополнительные функции и улучшив визуализацию данных для улучшения пользовательского опыта.

Что нужно сделать:

- Реализуй отображение маршрута с прогнозами погоды на интерактивной карте.
- Добавь дополнительные возможности для интерактивного взаимодействия с графиками (например, изменение временных интервалов, фильтрация по параметрам).
- Улучшай визуализацию, добавляя всплывающие подсказки, легенды и другие элементы, которые помогут пользователю лучше понимать прогнозы погоды.

Критерии оценки:

- 1. Интерактивные карты маршрута — 6 баллов:**
 - Маршрут корректно отображается на интерактивной карте.
 - Погодные условия для каждой точки маршрута представлены на карте.
- 2. Улучшение визуализации графиков — 4 баллов:**
 - Графики интерактивны и позволяют пользователю лучше взаимодействовать с данными.
 - Добавлены дополнительные элементы визуализации (всплывающие подсказки, легенды и т.д.).

Максимальный балл за задание — **10 баллов**.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Какой результат выполнения следующего кода? <pre>x = 0 x += 1 x += 2 print(x)</pre>	3	ПК-3
2.	Какой результат выполнения следующего кода? <pre>print("Hello".upper())</pre>	HELLO	ПК-3
3.	Сколько раз выполнится следующий цикл? <pre>for i in range(2, 9, 2): print(i)</pre>	4	ПК-3
4.	Какой результат выполнения следующего кода? <pre>a = [1, 2, 3] print(a[1])</pre>	2	ПК-3
5.	Какой модуль стандартной библиотеки Python используется для генерации случайных чисел? А. random Б. math В. statistics Г. itertools	А	ПК-3
6.	Какой встроенной функцией можно определить количество элементов в списке? А. list() Б. len() В. slice() Г. sum()	Б	ПК-3
7.	Разработайте функцию divide_numbers(a, b), которая принимает два числа и возвращает результат их деления. Если происходит деление на ноль, функция должна вернуть строку "Деление на ноль". Формат ответа: Пропуск <pre>def divide_numbers(a, b): try: return a / b except ZeroDivisionError: _____ "Деление на ноль"</pre>	return	ПК-4
8.	Какой результат выполнения следующего кода?	[1, 2, 3, 4]	ПК-4

	<pre>a = [1, 2, 3] a.append(4) print(a)</pre>		
9.	<p>Какой результат выполнения следующего кода?</p> <pre>a = [1, 2, 3, 2, 4] print(a.count(2))</pre>	2	ПК-4
10.	<p>Какой результат выполнения следующего кода?</p> <pre>a = [10, 20, 30, 40] print(a[1:3])</pre>	[20, 30]/20,30/20.30/[20.30]	ПК-4
11.	<p>Какое ключевое слово используется для объявления класса в Python?</p> <p>A. def Б. class B. function Г. object</p>	Б	ПК-4
12.	<p>Какой блок используется для обработки исключений в Python?</p> <p>A. try/except Б. if/else B. for/while Г. do/watch</p>	А	ПК-4