

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Галунин Сергей Александрович  
Должность: проректор по учебной работе  
Дата подписания: 23.12.2025 13:42:26  
Уникальный программный ключ:  
08ef34338325bdb0ac5a47baa5472ce36cc3fc3b

Приложение к ОПОП  
«Разработка программно-  
информационных систем»



**СПбГЭТУ «ЛЭТИ»**  
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

МИНОБРНАУКИ РОССИИ

федеральное государственное автономное образовательное учреждение высшего образования  
**«Санкт-Петербургский государственный электротехнический университет  
«ЛЭТИ» им. В.И.Ульянова (Ленина)»  
(СПбГЭТУ «ЛЭТИ»)**

---

## **РАБОЧАЯ ПРОГРАММА**

дисциплины

**«ОСНОВЫ БЭКЕНД-РАЗРАБОТКИ НА PYTHON Ч.1»**

для подготовки бакалавров

по направлению

09.03.04 «Программная инженерия»

по профилю

**«Разработка программно-информационных систем»**

Санкт-Петербург

2025

## **ЛИСТ СОГЛАСОВАНИЯ**

Разработчики:

доцент, к.т.н., доцент Заславский М.М.

ассистент Иванов Д.В.

Рабочая программа рассмотрена и одобрена на заседании кафедры МОЭВМ  
20.01.2025, протокол № 1

Рабочая программа рассмотрена и одобрена учебно-методической комиссией  
ФКТИ, 28.01.2025, протокол № 1

Согласовано в ИС ИОТ

Начальник ОМОЛА Загороднюк О.В.

## **1 СТРУКТУРА ДИСЦИПЛИНЫ**

Обеспечивающий факультет	ФКТИ
Обеспечивающая кафедра	МОЭВМ
Общая трудоемкость (ЗЕТ)	3
Курс	3
Семестр	5

### **Виды занятий**

Электронные лекции (акад. часов)	34
Электронные практические (академ. часов) (академ. часов)	34
Иная контактная работа (академ. часов)	1
Все контактные часы (академ. часов)	1
Самостоятельная работа, включая часы на контроль (академ. часов)	39
Всего (академ. часов)	108

### **Вид промежуточной аттестации**

Дифф. зачет (курс)	3
--------------------	---

## **2 АННОТАЦИЯ ДИСЦИПЛИНЫ**

### **«ОСНОВЫ БЭКЕНД-РАЗРАБОТКИ НА PYTHON Ч.1»**

Дисциплина является первой частью курса, в рамках которого обучающиеся получают знания, умения и навыки программирования на языке Python, изучают основы работы с IDE, получают навыки работы с базами данных и важнейшими библиотеками. Это позволит обучающимся писать скрипты для парсинга и автоматизации задач, работать с данными и создавать приложения на Python.

#### **SUBJECT SUMMARY**

#### **«BASICS OF PYTHON BACKEND DEVELOPMENT PART 1»**

The discipline is the first part of the course, in which students gain knowledge, skills, and programming skills in Python, learn the basics of working with the IDE, and gain skills in working with databases and critical libraries. This will allow students to write scripts for parsing and automating tasks, work with data, and create applications in Python.

## **3 ОБЩИЕ ПОЛОЖЕНИЯ**

### **3.1 Цели и задачи дисциплины**

1. Целью дисциплины является получение теоретических знаний в области разработки приложений для мобильных платформ, а также формирование практических умений и навыков по применению полученных знаний для решения задач профессиональной деятельности.
2. Задачами дисциплины являются приобретение знаний, умений и навыков в области бэкенд-разработки на языке Python.
3. Обучающиеся получают знания об объектно-ориентированном программировании при разработке бэкенда, технологиях создания проектов на Django, работы с путями и view-функциями, об основах вёрстки для бэкенда и использовании HTML-шаблонов.
4. В результате освоения дисциплины приобретаются умения работы с базами данных, построения моделей Django ORM, работы с формами и представлениями.
5. Обучающиеся приобретают навыки разработки базы данных для учебного проекта, использования библиотек unittest и pytest в рамках тестирования на Python.

### **3.2 Место дисциплины в структуре ОПОП**

Дисциплина изучается на основе ранее освоенных дисциплин учебного плана:

1. «Информатика»
2. «Программирование»
3. «Информационные технологии»
4. «Алгоритмы и структуры данных»

и обеспечивает изучение последующих дисциплин:

1. «Основы бэкенд-разработки на Python ч.2»
2. «Производственная практика (преддипломная практика)»

### **3.3 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы**

В результате освоения образовательной программы обучающийся должен достичь следующие результаты обучения по дисциплине:

<b>Код компетенции/ индикатора компетенции</b>	<b>Наименование компетенции/индикатора компетенции</b>
ПК-0	Способен разрабатывать информационные модели и применять их для решения задач профессиональной деятельности
ПК-0.1	<i>Знает современные виды информационных моделей, применяемых при решении задач профессиональной деятельности</i>
ПК-0.2	<i>Создает и модифицирует информационные модели для решения задач профессиональной деятельности</i>
ПК-0.3	<i>Применяет информационные модели для решения задач профессиональной деятельности</i>

## **4 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ**

### **4.1 Содержание разделов дисциплины**

#### **4.1.1 Наименование тем и часы на все виды нагрузки**

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>ЭЛек, ач</b>	<b>ЭПр, ач</b>	<b>ИКР, ач</b>	<b>СР, ач</b>
1	Введение	1			
2	Знакомство с Python	5	5		5
3	Основы Python	5	5		5
4	Объектно-ориентированное программирование при разработке бэкенда	5	6		6
5	Создание проекта	5	6		6
6	Базы данных	6	6		6
7	Тестирование на Python	6	6		6
8	Заключение	1		1	5
	Итого, ач	34	34	1	39
	Из них ач на контроль	0	0	0	0
	Общая трудоемкость освоения, ач/зе				108/3

#### **4.1.2 Содержание**

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>Содержание</b>
1	Введение	Введение в курс. Рабочая программа, коммуникации между заинтересованными лицами
2	Знакомство с Python	Переменные и типы. Циклы и ветвления. Функции. Словари и множества. Строки и форматирование. Библиотеки. Протокол HTTP. Сетевые запросы
3	Основы Python	Погружение в Python. Простые типы данных. Ветвления. Импортируемые типы данных. Коллекции. Итерации и циклы.
4	Объектно-ориентированное программирование при разработке бэкенда	Настройка рабочего окружения. Система контроля и управления версиями. Требования к коду. Отладка программ. Объекты и классы. Знакомство с ООП. Python: новый уровень. Расширенные возможности Python
5	Создание проекта	Фреймворк Django. Пути и view-функции. Вёрстка для бэкендера. HTML и шаблоны. Django. Проект

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>Содержание</b>
6	Базы данных	Введение в базы данных. Отношения между таблицами. Django ORM. Модели. Админ-зона Django. Django ORM. Получение информации из БД. Работа с формами. Представления: расширенные возможности. Пользователи в Django
7	Тестирование на Python	Тестирование. Библиотека unittest. Unittest в Django. Библиотека pytest. Pytest для Django
8	Заключение	Общие выводы по изложенному материалу. Содержание второй части курса. Связь с последующими дисциплинами траектории.

#### **4.2 Перечень лабораторных работ**

Лабораторные работы не предусмотрены.

#### **4.3 Перечень практических занятий**

<b>Наименование практических занятий</b>	<b>Количество ауд. часов</b>
1. Создание проекта	10
2. Базы данных	12
3. Тестирование на Python	12
Итого	34

#### **4.4 Курсовое проектирование**

Курсовая работа (проект) не предусмотрены.

#### **4.5 Реферат**

Реферат не предусмотрен.

#### **4.6 Индивидуальное домашнее задание**

Индивидуальное домашнее задание не предусмотрено.

#### **4.7 Доклад**

Доклад не предусмотрен.

## **4.8 Кейс**

Кейс не предусмотрен.

## **4.9 Организация и учебно-методическое обеспечение самостоятельной работы**

Самостоятельной записи на курс нет. Студент заходит на курс, используя логин/пароль от единой учетной записи университета (единий логин и пароль). Каждую неделю будет доступна новая тема курса: кратко раскрывающие содержание каждой темы, презентации и конспекты, с которыми обучающиеся смогут ознакомиться в любое удобное время. Все темы включают практические занятия, которые предусматривают самостоятельное выполнение заданий, а также задания с автоматической проверкой, результаты которых учитываются при общей аттестации полученных знаний. В конце каждой лекции необходимо пройти небольшой контрольный тест, который покажет насколько усвоен предложенный материал. Рекомендуем изучать материал последовательно, что существенно облегчит работу. У каждого контрольного задания имеется своя форма (тест или практическое задание) есть срок выполнения (окончательный срок), по истечении которого даже правильные ответы система принимать не будет! В расписании курса указан окончательный срок каждого задания, который варьируется от двух до четырех недель в зависимости от его сложности. Весь учебный курс рассчитан на 16 недель. Его итоги будут подведены в течение нескольких недель после его окончания.

<b>Текущая СРС</b>	<b>Примерная трудоемкость, ач</b>
Работа с лекционным материалом, с учебной литературой	10
Опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	6
Самостоятельное изучение разделов дисциплины	0
Выполнение домашних заданий, домашних контрольных работ	0
Подготовка к лабораторным работам, к практическим и семинарским занятиям	15

<b>Текущая СРС</b>	<b>Примерная трудоемкость, ач</b>
Подготовка к контрольным работам, коллоквиумам	0
Выполнение расчетно-графических работ	0
Выполнение курсового проекта или курсовой работы	0
Поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	0
Работа над междисциплинарным проектом	0
Анализ данных по заданной теме, выполнение расчетов, составление схем и моделей, на основе собранных данных	0
Подготовка к зачету, дифференцированному зачету, экзамену	8
<b>ИТОГО СРС</b>	<b>39</b>

## **5 Учебно-методическое обеспечение дисциплины**

### **5.1 Перечень основной и дополнительной литературы, необходимой для освоения дисциплины**

<b>№ п/п</b>	<b>Название, библиографическое описание</b>	<b>К-во экз. в библ.</b>
<b>Основная литература</b>		
1	Баланов А. Н. Бэкенд-разработка веб-приложений: архитектура, проектирование и управление проектами : учебное пособие для вузов / А. Н. Баланов, 2024. -312 с. -Текст : электронный.	неогр.
2	Чернышев С. А. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА PYTHON 2-е изд., пер. и доп. : Учебник для СПО / А. Чернышев С., 2025. -349 с. -Текст : непосредственный.	неогр.
<b>Дополнительная литература</b>		
1	Янцев В. В. Web-программирование на Python : учебное пособие для вузов / В. В. Янцев, 2024. -180 с. -Текст : электронный.	неогр.
2	Информатика. Введение в Python : электрон. учеб. пособие / К. В. Кринкин [и др.], 2020. -1 эл. опт. диск (CD-ROM). -Текст : электронный.	неогр.

### **5.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», используемых при освоении дисциплины**

<b>№ п/п</b>	<b>Электронный адрес</b>
1	Руководство по работе Docker в Python <a href="https://uproger.com/gajd-po-rabote-docker-v-python-ot-ustanovki-do-slozhnyh-primerov/">https://uproger.com/gajd-po-rabote-docker-v-p</a>

### **5.3 Адрес сайта курса**

Адрес сайта курса: <https://vec.etu.ru/moodle/course/view.php?id=23632>

## **6 Критерии оценивания и оценочные материалы**

### **6.1 Критерии оценивания**

Для дисциплины «Основы бэкенд-разработки на Python ч.1» формой промежуточной аттестации является дифф. зачет. Оценивание качества освоения дисциплины производится с использованием рейтинговой системы.

#### **Дифференцированный зачет**

<b>Оценка</b>	<b>Количество баллов</b>	<b>Описание</b>
Неудовлетворительно	0 – 7	теоретическое содержание курса не освоено, необходимые практически навыки и умения не сформированы, выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над курсом не приведет к существенному повышению качества выполнения учебных заданий
Удовлетворительно	8 – 11	теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки и умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки
Хорошо	12 – 13	теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки и умения сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками
Отлично	14 – 15	теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки и умения сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено количеством баллов, близким к максимальному

## **Особенности допуска**

К дифференцированному зачету допускаются студенты, показавшие прогресс в ЭОС Moodle (не менее 80%) и выполнившие 3 практические работы на оценку не менее "удовлетворительно".

Дифференцированный зачет проводится в виде тестирования в электронной образовательной среде Moodle. Тест содержит 15 вопросов с развернутыми ответами. Каждый вопрос теста оценивается в 1 балл в случае правильного ответа. Оценка дифференцированного зачета выставляется в зависимости от количества баллов, полученных за итоговый тест следующим образом:

"Неудовлетворительно" -менее 8 баллов

"Удовлетворительно" -от 8 до 11 баллов

"Хорошо" -от 12 до 13 баллов

"Отлично" -от 14 до 15 баллов.

## **6.2 Оценочные материалы для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине**

### **Вопросы к дифф.зачету**

<b>№ п/п</b>	<b>Описание</b>
1	Переменные и типы.
2	Циклы и ветвления
3	Функции.
4	Словари и множества
5	Строки и форматирование
6	Библиотеки.
7	Простые типы данных.
8	Ветвления.
9	Импортируемые типы данных
10	Система контроля и управления версиями
11	Требования к коду
12	Python: новый уровень. Расширенные возможности Python
13	Фреймворк Django.
14	Вёрстка для бэкендера
15	Введение в базы данных.
16	Django ORM

17	Админ-зона Django
18	Получение информации из БД
19	Тестирование. Библиотека unittest.
20	Unittest в Django. Библиотека pytest.

## Вариант теста

1. Какой метод HTTP используется для получения данных от сервера?
- POST
  - GET
  - PUT
  - DELETE

Ответ: GET

2. Какой статус HTTP означает «Не найдено»?
- 200
  - 404
  - 500
  - 302

Ответ: 404

3. Какой метод библиотеки requests используется для отправки GET-запроса?
- requests.post()
  - requests.get()
  - requests.send()
  - requests.fetch()

Ответ: requests.get()

4. Какой заголовок HTTP-ответа указывает тип содержимого?
- Content-Type
  - Accept
  - Content-Length

- Encoding

Ответ: Content-Type

5. Какой модуль Python используется для кодирования URL?

- urllib.parse
- requests.utils
- http.encode
- url.encode

Ответ: urllib.parse

6. Что такое модули в Node.js?

- Настройки приложения
- Файлы, содержащие функции и переменные, которые могут быть повторно использованы
- Команды командной строки
- Файлы с базами данных

Ответ: Файлы, содержащие функции и переменные, которые могут быть повторно использованы

7. Какой механизм обычно используется для аутентификации пользователей в веб-приложениях?

- HTTPS
- JWT (JSON Web Token)
- CSS
- SQL

Ответ: JWT (JSON Web Token)

8. Какой из следующих форматов данных является наиболее популярным для передачи данных между клиентом и сервером?

- XML

- JSON
- CSV
- YAML

Ответ: JSON

9. Какой из следующих типов баз данных чаще всего используется для хранения временных рядов данных?

- Реляционная база данных
- Документо-ориентированная база данных
- Графовая база данных
- Time-series база данных

Ответ: Time-series база данных

### **Образцы задач (заданий) для контрольных (проверочных) работ**

#### **Пример практической работы**

Напишите программу, которая будет:

- хранить перечень продуктов, их количество и срок годности;
- позволит добавлять новые продукты в перечень продуктов;
- позволит получать информацию о содержимом холодильника.

#### *Пояснения к выполнению задания*

```
““python
goods = {
    ’Пельмени Универсальные’: [
        # Первая партия продукта ’Пельмени Универсальные’:
        {’amount’: Decimal(’0.5’), ’expiration_date’: datetime.date(2023, 7, 15)},
        # Вторая партия продукта ’Пельмени Универсальные’:
        {’amount’: Decimal(’2’), ’expiration_date’: datetime.date(2023, 8, 1)},
```

```
],  
    'Вода': [  
        {'amount': Decimal('1.5'), 'expiration_date': None}  
    ],  
}  
""
```

В примере видно, что продукт «Вода» закуплен один раз в объёме 1.5 кг. Однако продукт с одним и тем же названием может быть закуплен и добавлен в холодильник несколько раз.

Например, неделю назад пользователь купил килограмм пельменей под названием «Универсальные», половину съел, а потом пошёл и купил ещё два килограмма. Название у продукта одно и то же, но срок хранения может оказаться разным. Каждую такую закупку будем называть партия продукта.

Таким образом, у любителя пельменей в холодильнике окажется две партии продукта «Пельмени Универсальные». В первой партии будет 0.5 кг продукта, во второй — 2 кг. Срок хранения у этих партий тоже будет разным.

Ключ каждого элемента словаря — это название определённого продукта. Значение каждого элемента словаря — это список словарей.

Каждый вложенный словарь содержит два элемента:

`amount` — количество продукта.

`expiration_date` — срок годности.

Количество продукта — это:

целое число для исчисляемых продуктов (в штуках: например, значение 10 для яиц);

дробное число для весовых продуктов (в килограммах: например, значе-

ние 1.5 для воды).

Если срок годности не указан, вместо него должно быть указано None.

В программе должно быть пять функций для управления содержимым холодильника:

`add()` — добавляет продукт в словарь `goods`;

`add_by_note()` — добавляет продукт в словарь `goods`, преобразуя текстовое описание продукта в структурированные данные;

`find()` — ищет в словаре `goods` заданное слово или строку и возвращает список продуктов, в названии которых есть это слово;

`amount()` — возвращает количество запрошенного продукта;

Весь комплект контрольно-измерительных материалов для проверки сформированности компетенции (индикатора компетенции) размещен в закрытой части по адресу, указанному в п. 5.3

### **6.3 График текущего контроля успеваемости**

<b>Неделя</b>	<b>Темы занятий</b>	<b>Вид контроля</b>
1	Знакомство с Python Основы Python	
2		
3		
4		
5		Практическая работа
6	Объектно-ориентированное программирование при разработке бэкенда Создание проекта	
7		
8		
9		
10		
11	Базы данных Тестирование на Python	Практическая работа
12		
13		
14		
15		
16		Практическая работа

### **6.4 Методика текущего контроля**

#### **на лекционных занятиях**

Текущий контроль включает в себя:

- контроль прогресса в ЭОС Moodle (не менее 80%).

#### **на практических занятиях**

- контроль прогресса в ЭОС Moodle (не менее 80%)

В процессе обучения по дисциплине студент обязан выполнить 3 практические работы в ЭОС Moodle.

Студент должен показать: понимание постановки задачи, подхода к ее решению, ход решения, выбор тех или иных методик решения задачи. Преподаватель определяет глубину понимания теоретического материала, который лежит в основе решения задачи практической работы, а также самостоятельность ее выполнения.

**Критерии оценивания практических работ:**

«отлично» - вопрос раскрыт полностью, задача решена правильно

«хорошо» - вопрос раскрыт не полностью, задача решена частично

«удовлетворительно» - в ответе на вопрос имеются существенные ошибки; задача не решена или решена неправильно, ход решения правильный

«неудовлетворительно» - отсутствует ответ на вопрос или содержание ответа не совпадает с поставленным вопросом, задача не решена, ход решения неправильный

### **самостоятельной работы студентов**

Контроль самостоятельной работы студентов осуществляется на осуществляется на лекционных, практических занятиях по методикам, описанным выше.

## **7 Описание информационных технологий и материально-технической базы**

<b>Тип занятий</b>	<b>Тип помещения</b>	<b>Требования к помещению</b>	<b>Требования к программному обеспечению</b>
Лекция	Лекционная аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, ПК	1) Linux Альт Образование 10 и выше; 2) Р7-Офис 7 и выше либо LibreOffice 7 и выше
Практические занятия	Аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, ПК	1) Linux Альт Образование 10 и выше; 2) Р7-Офис 7 и выше либо LibreOffice 7 и выше
Самостоятельная работа	Помещение для самостоятельной работы	Оснащено компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.	1) Linux Альт Образование 10 и выше; 2) Р7-Офис 7 и выше либо LibreOffice 7 и выше

## **8 Адаптация рабочей программы для лиц с ОВЗ**

Адаптированная программа разрабатывается при наличии заявления со стороны обучающегося (родителей, законных представителей) и медицинских показаний (рекомендациями психолого-медико-педагогической комиссии). Для инвалидов адаптированная образовательная программа разрабатывается в соответствии с индивидуальной программой реабилитации.

## **ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

<b>№ п/п</b>	<b>Дата</b>	<b>Изменение</b>	<b>Дата и номер протокола заседания УМК</b>	<b>Автор</b>	<b>Начальник ОМОЛА</b>