

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
САМАРСКИЙ УНИВЕРСИТЕТ ГОСУДАРСТВЕННОГО УПРАВЛЕНИЯ  
«МЕЖДУНАРОДНЫЙ ИНСТИТУТ РЫНКА»

УТВЕРЖДАЮ  
Проректор по учебной работе и  
качеству образования

\_\_\_\_\_ И. А. Долгова

15 апреля 2026 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЫ

**РАЗРАБОТКА КОРПОРАТИВНЫХ ИНТЕРНЕТ-ПОРТАЛОВ**

---

Направление подготовки:	09.03.03 Прикладная информатика
Профиль подготовки:	Корпоративные информационные системы
Квалификация:	бакалавр
Форма обучения:	очно-заочная
Год начала подготовки:	2026

Самара  
2026

# 1. ОЦЕНОЧНЫЕ СРЕДСТВА, СООТНЕСЁННЫЕ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Код и наименование универсальной компетенции	Код и наименование индикатора достижения универсальной компетенции	Результаты обучения по дисциплинам и практикам	Вид аттестации и оценочных средств
ПК-3. Способен проектировать базы данных, а также кодировать на современных языках программирования клиентскую и серверную части информационной системы	ПК-3.И-1. Разрабатывает базы данных для корпоративных информационных систем	ПК-3.И-1.3-1. Знает основные типы баз данных, их архитектуру и системы управления	Текущий контроль: промежуточный тест, лабораторная работа. Промежуточная аттестация: вопросы к экзамену
		ПК-3.И-1.У-1. Умеет проектировать базы данных для информационных систем различного назначения	
	ПК-3.И-2. Разрабатывает клиентскую и серверные части информационных систем с использованием соответствующих языков программирования	ПК-3.И-2.3-1. Знает основные методы front-end разработки	Текущий контроль: промежуточный тест, лабораторная работа. Промежуточная аттестация: вопросы к экзамену
		ПК-3.И-2.У-1. Умеет применять основные методы front-end разработки	
		ПК-3.И-2.3-2. Знает основные методы back-end разработки	
		ПК-3.И-2.У-2. Умеет применять основные методы back-end разработки	

## 2. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ

### 2.1. Темы лабораторных работ

Раздел 1. Введение во фреймворки для разработки back-end приложений.

Лабораторная работа №1. Знакомство с back-end фреймворками.

*Цель работы:* Ознакомиться с общими принципами работы фреймворков для разработки серверных приложений.

*Вопросы для самопроверки:*

1. Что такое Web-приложение?
2. Из каких частей состоит Web-приложение?
3. Какую роль в Web-приложении играет серверная часть?
4. Какие back-end фреймворки поддерживают Python?
5. На каких языках программирования пишут программы на стороне сервера?

Раздел 2. Введение во фреймворк Flask.

Лабораторная работа №2. Общая архитектура фреймворка.

*Цель работы:* Ознакомиться с принципами организации Flask.

*Вопросы для самопроверки:*

1. В чем преимущества Flask?
2. Что такое стандарт WSGI?
3. Как установить Flask в IDE PyCharm?
4. Что такое представление?
5. Как реализован механизм создания функции-обработчика запросов?

Раздел 3. Стилизация шаблонов.

Лабораторная работа №3. Использование Html –шаблонов в приложениях на Flask.

*Цель работы:* Научиться работать с шаблонами.

*Вопросы для самопроверки:*

1. Зачем нужны Html – шаблоны в приложениях на Flask?
2. Как создать базовый шаблон?
3. Как на основе базового шаблона создать шаблон-потомок?
4. Как стилизовать оформление шаблона?
5. Как реализовать вывод информации в шаблон по условию и циклу?

Раздел 4. Формы.

Лабораторная работа №4. Работа с Html-формами.

*Цель работы:* Научиться обрабатывать информацию из Html - форм.

*Вопросы для самопроверки:*

1. Что такое Html - форма?
2. Какими методами осуществляется передача на сервер данных из формы?
3. Каким требованиям должна удовлетворять функция обработчик?
4. Как при обработке запроса используется объект request.form?
5. Как прочитать информацию из радио-кнопок и чек-боксов?

Раздел 5. Базы данных.

Лабораторная работа №5. Работа с базой данных.

*Цель работы:* Научиться использовать базы данных в Web-приложениях .

*Вопросы для самопроверки:*

1. Как расшифровывается аббревиатура ORM?
2. Какие системы ORM Вы знаете?
3. В чем основные преимущества ORM технологии?
4. Как с помощью ORM подключиться к базе данных?
5. Как реализовать операции CRUD с помощью ORM?

## Раздел 6. Мгновенные сообщения.

Лабораторная работа №6. Мгновенные сообщения в приложениях на Flask.

*Цель работы:* Научиться создавать мгновенные сообщения.

*Вопросы для самопроверки:*

1. Что такое мгновенные сообщения?
2. В чем их особенности и преимущества?
3. Как можно стилизовать мгновенные сообщения?
4. Как используется функция flash() для реализации мгновенных сообщений?
5. Как используется функция get\_flashed\_messages() для реализации мгновенных сообщений?

## Раздел 7. Разработка приложения "Мотивационный тест".

Лабораторная работа №7. Реализация приложения "Мотивационный тест"

*Цель работы:* Получить навыки разработки конкретного приложения.

*Вопросы для самопроверки:*

1. Что из себя представляет мотивационный тест ?
2. Какие параметры у этого теста?
3. Каков алгоритм обработки теста?
4. Какова архитектура приложения?
5. В каком виде задаются входные параметры и сохраняются результаты?

## Раздел 8. Разработка приложения "Тест Томаса".

Лабораторная работа №8. Реализация приложения "Тест Томаса"

*Цель работы:* Получить навыки разработки конкретного приложения.

*Вопросы для самопроверки:*

1. Что из себя представляет тест Томаса?
2. Какие параметры у этого теста?
3. Каков алгоритм обработки теста?
4. Какова архитектура приложения?
5. В каком виде задаются входные параметры и сохраняются результаты?

## Раздел 9. Разработка приложения "Тест Кейрси".

Лабораторная работа №9. Реализация приложения "Тест Кейрси"

*Цель работы:* Получить навыки разработки конкретного приложения.

*Вопросы для самопроверки:*

1. Что из себя представляет тест Кейрси?
2. Какие параметры у этого теста?
3. Каков алгоритм обработки теста?
4. Какова архитектура приложения?
5. В каком виде задаются входные параметры и сохраняются результаты?

**Методические указания к проведению лабораторных работ**

**Цели лабораторных занятий:**

1. Углубление и закрепление знания теоретического курса путем практического изучения в лабораторных условиях изложенных в лекциях методов и технологий;
2. Приобретение навыков в научном экспериментировании, анализе полученных результатов;
3. Формирование первичных навыков организации, планирования и проведения научных исследований.

### **Порядок проведения лабораторного занятия:**

1. Вводная часть:
  - входной контроль подготовки обучаемого;
  - вводный инструктаж (знакомство обучающихся с содержанием предстоящей работы, краткий анализ теоретических положений и выводов, демонстрация подходов к выполнению отдельных операций, напоминание о технике безопасности, предупреждение о возможных ошибках).
2. Основная часть:
  - проведение обучаемым лабораторной работы;
  - текущее индивидуальное консультирование обучаемого;
3. Заключительная часть:
  - демонстрация результатов выполненного задания;
  - заключительный инструктаж (подведение итогов выполнения учебных задач, разбор допущенных ошибок и выявление их причин, сообщение результатов работы каждого обучаемого, объявление о том, что необходимо повторить к следующему занятию).

### **Особенности подготовки к проведению лабораторного занятия**

Подготовка лабораторного занятия начинается с изучения теоретических положений, определения (уточнения) целей и задач данного занятия, времени, выделяемого обучаемым для подготовки.

В ходе подготовки к лабораторной работе необходимо пояснить проблематику, объем и содержание лабораторного занятия, определить, какие понятия, определения, теории могут быть иллюстрированы данным экспериментом, какие умения и навыки должны приобрести обучаемые в ходе занятия, какие знания углубить и расширить.

При этом преподавателю необходимо решить, на каком этапе обучения следует поставить задачу о подготовке к лабораторной работе, каким образом достигнуть активизации познавательной деятельности обучающихся. Задача на подготовку к лабораторной работе может быть поставлена на лекции, с таким временным расчетом, чтобы обучаемые смогли качественно подготовиться к ее проведению. Одновременно им выдаются учебно-методические материалы, иллюстрирующие круг вопросов, затрагиваемых в ходе выполнения лабораторного задания. Это могут быть методические указания по соответствующему курсу, презентации, ссылки на Интернет-источники и др. Эти материалы могут отражать учебные вопросы, краткие сведения по теории, программу выполнения работы, содержание отчета, вопросы для подготовки и литературу, рекомендуемую к изучению и т.д. В них также ставятся задачи, которые обучаемые должны решить при подготовке к работе, в процессе эксперимента и при обработке полученных результатов.

В ходе подготовки к лабораторной работе необходимо обратить внимание обучающегося на результат ее выполнения. Результат лабораторной работы должен быть четко сформулирован, приведены критерии его достижения, перечень материалов, его (результат) иллюстрирующих – файлы, графики, скриншоты и т.д. Учащийся должен уметь формулировать основные выводы, опираясь на полученный на лабораторной работе результат.

В отдельных случаях, на лабораторном занятии может быть предусмотрена защита выполненной работы.

### **Шкала и критерии оценки лабораторной работы**

Критерии	Баллы
----------	-------

Критерии	Баллы
Степень соответствия выполненного задания поставленным требованиям	25
Структурирование и комментирование лабораторной работы	25
Уникальность выполненной работы (отличие от работ коллег)	25
Ответы на контрольные вопросы	25

Лабораторная работа оценивается по 100 балльной шкале, баллы переводятся в оценки успеваемости следующим образом:

90 – 100 баллов – «отлично»;

70 – 89 баллов – «хорошо»;

50 – 69 баллов – «удовлетворительно»;

менее 50 баллов – «неудовлетворительно».

### 3. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

#### 3.1. Банк контрольных заданий (с указанием компетенции)

##### 1. Прочитайте текст и установите соответствие. (ПК-3.1)

Установите соответствие между языками программирования (1-4) и back-end фреймворками (А-Г):

1. Java	А. Django
2. PHP	Б. Node.js
3. Python	В. Spring
4. Javascript	Г. Laravel

А	Б	В	Г

##### 2. Прочитайте текст и установите последовательность (ПК-3.1)

Установите правильную последовательность операторов языка программирования Python с использованием фреймворка Flask для реализации вывода строки "Hello, World" в браузер

- А. `def index():`  
     `return ("<h1> Hello, World </h1>")`  
 Б. `@app.route("/")`  
 В. `app=Flask(__name__)`  
 Г. `if __name__=="main":`  
     `app.run(debug=True)`  
 Д. `from flask import Flask`

--

##### 3. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ПК-3.1)

Где должны храниться Html-шаблоны при разработке приложений на Flask :

- А. В произвольной папке  
 Б. В произвольной папке проекта  
 В. В папке templates проекта  
 Г. В папке renders проекта  
 Д. В папке static проекта


##### 4. Прочитайте текст и запишите развёрнутый обоснованный ответ. (ПК-3.1)

В чем основное преимущество использования фреймворков при разработке Web-приложений

--

**5. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ПК-3.1)**

Где должны храниться css файлы при разработке приложений на Flask :

- А. В произвольной папке
- Б. В произвольной папке проекта
- В. В папке templates проекта
- Г. В папке renders проекта
- Д. В папке static проекта


**6. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ПК-3.1)**

Какое действие выполняет следующий код в приложении Flask :

```
<ul>
{% for m in menu %}
<li> {{m}} </li>
{% endfor %}
</ul>
```

- А. Формирует запрос к базе данных
- Б. Создает Html-шаблон
- В. Выводит в Html шаблон данные, сформированные сервером
- Г. Другой ответ


**7. Прочитайте текст и установите соответствие. (ПК-3.1)**

Установите соответствие между элементами программной конструкции (1-3) и их назначением (А-В):

<pre>&lt;ul&gt; {% for m in menu %} &lt;li&gt; {{m}} &lt;/li&gt; {% endfor %} &lt;/ul&gt;</pre>	
1. <b>m</b>	А. список, сформированный сервером
2. <b>menu</b>	Б. тег шаблона
3. <b>ul</b>	В. элемент списка

А	Б	В

**8. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ПК-3.2)**

Что означает выражение <string:name> в декораторе приложения Flask:

**@app.route("/user/<string:name>")**

- 1) позволяет из url адреса получить дополнительную строку name
- 2) имя пользователя
- 3) имя Html шаблона
- 4) ошибка

Ответ
Основание

**9. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ПК-3.2)**

С помощью какой функции во Flask осуществляется передача данных с сервера в Html-шаблон :

- 1) **return**
- 2) **app.route**
- 3) **render\_template**
- 4) **url\_for**

Ответ
Основание

**10. Прочитайте текст и установите соответствие. (ПК-3.2)**

Установите соответствие между именами элементов фреймворка Flask (1-4) и производимым ими действием (А-Г):

1. <b>@app.route</b>	А. функция для передачи ответа сервера
2. <b>url_for</b>	Б. декоратор для функции обработчика
3. <b>render_template</b>	В. список для чтения переданной информации из формы
4. <b>request_form</b>	Г. функция для доступа к файлам папки static

А	Б	В	Г

**11. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ПК-3.2)**

Какая функция во Flask обеспечивает доступ к CSS файлам проекта :

- 1) **return**
- 2) **app.route**
- 3) **render\_template**
- 4) **url\_for**

Ответ
-------

Основание
-----------

## 12. Прочитайте текст и установите соответствие. (ПК-3.2)

Установите соответствие между элементами Flask (1-4) и их назначением (А-Г):

1. <b>g</b>	А. переменная, содержащая данные о текущем запросе
2. <b>current_app</b>	Б. словарь для сохранения данных на время сеанса
3. <b>request</b>	В. глобальная переменная для сохранения пользовательской информации
4. <b>session</b>	Г. ссылка на контекст приложения

А	Б	В	Г

## 13. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ПК-3.2)

Какую роль играет объект **request** в приложениях на Flask:

- 1) обеспечивает доступ к html-шаблонам
- 2) обеспечивает доступ к css файла проекта и файлам изображений
- 3) позволяет получить доступ к данным, отправленным пользователем через формы и URL-адреса
- 4) другой ответ

## 14. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ПК-3.2)

С помощью какого метода обрабатывается информация на стороне сервера о выбранных чек-боксах:

- 1) input
- 2) url\_for
- 3) getlist
- 4) getter
- 5) другой ответ

Ответ
-------

## 15. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ПК-3.2)

Что представляет собой объект **request.form** в приложениях Flask

- 1) строку с передаваемыми данными
- 2) словарь с возвращаемой сервером информацией
- 3) строку с url адресом
- 4) другой ответ

## 16. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ПК-3.2)

Для чего используется элемент `<path:username>` в декораторе приложения Flask:  
`@app.route("/user/<path:username>")`

- 1) для улучшения читаемости программы
- 2) для задания пути к css файлам
- 3) для получения последней части строки url адреса
- 4) ошибка

Ответ
Основание

### 3.2. Ключи к контрольным заданиям

№ задания	Верный ответ
1	A3B4B1Г2
2	ДВБАГ
3	В В проектах на Flask Html-шаблоны должны храниться в папке templates
4	При использовании фреймворков имеются следующие преимущества: 1. Структура приложения заранее задана и сохраняется единой для любого приложения 2. Часть рутинных функций приложения берет на себя фреймворк, что значительно упрощает разработку
5	Д В проектах на Flask css файлы должны храниться в папке static
6	В Сервер формирует список menu и его элементы в цикле выводятся в шаблон
7	A2B3B1
8	1) Это конвертер Flask типа string
9	3) Функция render_template предназначена для передачи значений переменных, сформированных сервером в шаблон
10	A3B1B4Г2
11	4) Функция url_for предназначена для доступа к содержимому папки static проекта на Flask
12	A2B4B1Г3
13	3) Объект request предназначен для получения данных переженных из формы методом POST
14	3) Поскольку на стороне клиента могут быть выбраны несколько чек-боксов, их значения сохраняются в списке, доступ к которому можно получить через метод getlist
15	4) объект request.form представляет собой список кортежей с именем поля формы и ее значением
16	3) Это конвертер Flask для получения "хвоста" url адреса

#### Шкала и критерии оценки текущего тестирования

Число правильных ответов	Оценка
90-100% правильных ответов	Оценка «отлично»
70-89% правильных ответов	Оценка «хорошо»
50-69% правильных ответов	Оценка «удовлетворительно»
Менее 50% правильных ответов	Оценка «неудовлетворительно»

### 3.3. Перечень тем для проверки образовательных результатов на знания (вопросы к экзамену)

1. Понятие о корпоративных порталах и методах их разработки.
2. Обзор back-end фреймворков для различных языков программирования: Laravel, Spring, Django, Node.js и др.
3. Достоинства и недостатки различных фреймворков.
4. Сравнительный анализ Django и Flask – фреймворков для языка программирования Python.
5. Достоинства фреймворка Flask.
6. Стандарт WSGI.
7. Установка Flask в среде PyCharm.
8. Эмуляция локального сервера в среде PyCharm.
9. Формирование представления и функции-обработчика запроса – декоратор `@app.route()`.
10. Применение одного обработчика к нескольким url-адресам.
11. Формирование отклика сервера, понятие о шаблонах.
12. Размещение шаблонов в проекте.
13. Папка templates.
14. Передача параметров в шаблон.
15. Условие и цикл в шаблоне.
16. Механизм наследования базового шаблона (расширение шаблона).
17. Блоки базового шаблона.
18. Подключение стилей к шаблону – папка static.
19. Использование функции `url_for()` для подключения файлов css, рисунков.
20. Получение параметров запроса из url адреса.
21. Пример разработки простого приложения с использованием фреймворка Flask.
22. Использование Html-форм в проектах на Flask.
23. Передача данных из формы методами GET и POST.
24. Создание функции – обработчика формы.
25. Объект `request.form` и функция `render_template()`.
26. Обработка радиокнопок и чек-боксов.
27. Разработка приложения – теста на знание основ html с подключением базы данных SQLite.
28. Особенности использования баз данных в приложениях на Flask.
29. Примеры проектов.
30. Работа с базами данных SQLite, MySQL, PostgreSQL, MongoDB.
31. Использование технологии ORM (Object-relationalmapper) при работе с базами данных.
32. Проект с авторизацией пользователя и сохранением его профиля в базе данных.
33. Механизм реализации запросов во Flask: переменные `g`, `Current_app`, `request`, `session`.
34. Обслуживание нескольких запросов через контекст приложения.
35. Понятие мгновенного сообщения.
36. Функции `flash()` и `get_flashed_messages()`.
37. Применение стилей к мгновенным сообщениям.
38. Пример приложения с использованием мгновенных сообщений.
39. Разработка общей архитектуры проекта "Мотивационный тест".
40. Разработка клиентской части: выбор фреймворка, генерация основного и вспомогательного меню, стилизация главной и вспомогательных страниц, стили отображения входной и выходной информации.
41. Разработка серверной части: выбор базы данных, наполнение ее таблиц, разработка функций-обработчиков пунктов главного и вспомогательного меню и др.
42. Отладка проекта и размещение его на хостинге.
43. Разработка общей архитектуры проекта "Тест Томаса".

44. Разработка клиентской части: выбор фреймворка, генерация основного и вспомогательного меню, стилизация главной и вспомогательных страниц, стили отображения входной и выходной информации.
45. Разработка серверной части: выбор базы данных, наполнение ее таблиц, разработка функций-обработчиков пунктов главного и вспомогательного меню и др.
46. Отладка проекта и размещение его на хостинге.
47. Разработка общей архитектуры проекта "Тест Кейрси".
48. Разработка клиентской части: выбор фреймворка, генерация основного и вспомогательного меню, стилизация главной и вспомогательных страниц, стили отображения входной и выходной информации.
49. Разработка серверной части: выбор базы данных, наполнение ее таблиц, разработка функций-обработчиков пунктов главного и вспомогательного меню и др.
50. Отладка проекта и размещение его на хостинге.
51. Объединение 3 проектов в один и формирование портала "Социально-психологическое тестирование".