

Документ подписан простой электронной подписью Информация о владельце: ФИО: Таскаев Сергей Валерьевич Должность: Ректор Дата подписания: 25.06.2026 13:02:32	МИНОБРНАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)
Фонд оценочных средств для промежуточной аттестации по дисциплине "Разработка серверных приложений на языке Python" по направлению подготовки (специальности) "09.03.04 Программная инженерия" направленности (профилю) "Разработка программно-информационных систем ФГБОУ ВО «ЧелГУ»"	стр. 1

**Фонд оценочных средств для промежуточной аттестации по дисциплине
Разработка серверных приложений на языке Python**

Направление подготовки (специальность)

09.03.04 Программная инженерия

Направленность (профиль)

Разработка программно-информационных систем

Присваиваемая квалификация (степень)

бакалавр

Форма обучения

заочная форма обучения

Год(ы) набора 2026

Челябинск 2026 г.

09.03.04 Программная инженерия профиль Разработка программно-информационных систем, дисциплина Разработка серверных приложений на языке Python, 2026 год набора, заочная форма обучения

Фонд оценочных средств дисциплины (модуля) одобрен и рекомендован:

Проректор по учебной работе утверждено 27.02.2026 А.А. Саламатов

Ученым советом института информационных технологий

Протокол заседания № 7 от 26.02.2026

Председатель Ученого совета
института информационных
технологий

согласовано

Ю.В. Петриченко

Заседанием кафедры информационных технологий и экономической информатики

Протокол заседания №7 от 26.02.2026

Заведующий кафедрой

согласовано

С.А. Скрипов

Автор (составитель)

И.Е. Николаев

Структура фондов оценочных средств соответствует приказу ректора ФГБОУ ВО «ЧелГУ» от 27 сентября 2022 № 573-1



Содержание

1. Паспорт фонда оценочных средств	3
2. Перечень формируемых компетенций	4
3. Содержание оценочных средств по дисциплине	5
3.1. Виды оценочных средств	5
3.2. Содержание оценочных средств	6
4. Порядок проведения и критерии оценивания промежуточной аттестации	15
4.1. Порядок проведения промежуточной аттестации	15
4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств	15
4.3. Результаты промежуточной аттестации и уровни сформированности компетенций	15



1. Паспорт фонда оценочных средств

Направление подготовки: 09.03.04 Программная инженерия

Направленность: Разработка программно-информационных систем

Дисциплина: Разработка серверных приложений на языке Python.

Курс (курсы) изучения: 3

Форма промежуточной аттестации: зачёт

Для оценивания результатов обучения используется балльно-рейтинговая система.



2. Перечень формируемых компетенций

Изучение дисциплины «Разработка серверных приложений на языке Python» направлено на формирование компетенций, приведённых в 1.

Таблица 1. Результаты обучения по дисциплине.

Коды компетенции и согласно ФГОС (ОПОП ВО)	Содержание компетенций согласно ФГОС (ОПОП ВО)	Индикаторы достижения компетенции согласно ОПОП	Перечень планируемых результатов обучения по дисциплине
1	2	3	4
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1. Демонстрирует знание основ информатики, теории алгоритмов, методологии и технологии программирования ОПК-6.2. Демонстрирует умения разрабатывать алгоритмические и программные решения, проводить проектирование, конструирование и тестирование программных продуктов ОПК-6.3. Имеет практический опыт использования технологий разработки программного обеспечения	Знать: теорию алгоритмов, библиотеки и инструментальные средства для программирования на языке Python Уметь: разрабатывать информационные системы, системы автоматизации, программы для научных расчетов на языке Python Владеть: навыками использования технологий и библиотек для разработки программного обеспечения на языке Python
ПК-2	Владение навыками использования различных технологий промышленной разработки программного обеспечения с применением инструментов автоматизации сборки, интеграции, тестирования и развертывания ПО	ПК-2.1. Демонстрирует знание основных принципов и технологий промышленной разработки программного обеспечения ПК-2.2. Демонстрирует умения разрабатывать программное обеспечение с применением инструментов автоматизации сборки, интеграции, тестирования и развертывания ПО ПК-2.3. Имеет практический опыт промышленной разработки программного обеспечения	Знать: основные принципы и технологии объектно-ориентированного программирования на языке Python Уметь: использовать инструментальные средства для разработки, развертывания и тестирования программного обеспечения на языке Python Владеть: навыками разработки приложений в стиле объектно-ориентированного программирования на языке программирования Python навыками разработки приложений с внешними источниками данных навыками использования библиотек языка Python для научных вычислений и научной визуализации



3. Содержание оценочных средств по дисциплине

3.1. Виды оценочных средств

Таблица 2. Виды оценочных средств.

№ п/п	Код компетенции/ планируемые результаты обучения	Контролируемые темы/ разделы	Наименование оценочного средства для текущего контроля	Наименование оценочного средства на промежуточной аттестации/№ задания
1	ОПК-6.1. Демонстрирует знание основ информатики, теории алгоритмов, методологии и технологии программирования Знать: теорию алгоритмов, библиотеки и инструментальные средства для программирования на языке Python	Инструменты с Python. Синтаксис Python. Объектно-ориентированное программирование Разработка серверных приложений	Тест	Задания теста № 2-4, 12, 18, 23, 24, 39, 42, 50-55
2	ОПК-6.2. Демонстрирует умения разрабатывать алгоритмические и программные решения, проводить проектирование, конструирование и тестирование программных продуктов Уметь: разрабатывать информационные системы, системы автоматизации, программы для научных расчетов на языке Python	Инструменты с Python. Синтаксис Python. Объектно-ориентированное программирование Разработка серверных приложений	Тест	Задания теста № 1-55
3	ОПК-6.3. Имеет практический опыт использования технологий разработки программного обеспечения Владеть: навыками использования технологий и библиотек для разработки программного обеспечения на языке Python	Инструменты с Python. Синтаксис Python. Объектно-ориентированное программирование Разработка серверных приложений	Тест	Задания теста № 1-55
4	ПК-2.1. Демонстрирует знание основных принципов и технологий промышленной разработки программного обеспечения Знать: основные принципы и технологии объектно-ориентированного программирования на языке Python	Инструменты с Python. Синтаксис Python. Объектно-ориентированное программирование Разработка серверных приложений	Тест	Задания теста № 2-4, 12, 18, 23, 24, 39, 42, 50-55
5	ПК-2.2. Демонстрирует умения разрабатывать	Инструменты с Python. Синтаксис Python.	Тест	Задания теста № 1-55



	программное обеспечение с применением инструментов автоматизации сборки, интеграции, тестирования и развертывания ПО Уметь: использовать инструментальные средства для разработки, развертывания и тестирования программного обеспечения на языке Python	Объектно-ориентированное программирование Разработка серверных приложений		
6	ПК-2.3. Имеет практический опыт промышленной разработки программного обеспечения Владеть: навыками разработки приложений в стиле объектно-ориентированного программирования на языке программирования Python навыками разработки приложений с внешними источниками данных навыками использования библиотек языка Python для научных вычислений и научной визуализации	Инструменты с Python. Синтаксис Python. Объектно-ориентированное программирование Разработка серверных приложений	Тест	Задания теста № 1-55

Типовые задания, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе дисциплины (модуля). Полные комплекты оценочных средств и контрольно-измерительных материалов хранятся на кафедре.

3.2. Содержание оценочных средств

База тестовых вопросов

№ п/п	Формулировка вопроса	Варианты ответов (полужирным шрифтом – верные варианты)
1.	Какой вывод вы получите после выполнения кода?	a. helloNO b. helloZero Division Error c. NO d. hello



	<pre>def f(x): def f1(a, b): print("hello") if b==0: print("NO") return return f(a, b) return f1 @f def f(a, b): return a%b f(4,0)</pre>	
2.	<p>Какой код более производительный по времени?</p> <p>Первый: a = [x + 1 for x in range(100000000)] b = [x * 2 for x in a] c = [x ** 2 for x in a]</p> <p>Второй: a = (x + 1 for x in range(100000000)) b = (x * 2 for x in a) c = (x ** 2 for x in a)</p>	<p>a. Первый b. Второй</p>
3.	<p>Какой способ предпочтительнее?</p> <pre>with open("test.txt", encoding = 'utf-8') as f: # perform file operation try: f = open("test.txt", encoding = 'utf-8') # perform file operations finally: f.close()</pre>	<p>a. Первый b. Второй</p>
4.	<pre>b = {} b[0,0] = 1 b[0,1] = 2 b[0,2] = 3 print(b[0])</pre> <p>На какой строчке сломается код ниже:</p>	<p>a. 1 b. 2 c. 3 d. 4 e. 5</p>



5.	<pre>a = [[0]] a *= 3 a[0].append(1) print(a)</pre> <p>Что будет выведено в терминале</p>	<p>a. [[0, 1], [0, 1], [0, 1]] b. [[0, 0, 1], [0, 0, 1], [0, 0, 1]] c. [[0, 0, 0, 1]] d. [[0, 3], [0, 3], [0, 3]]</p>
6.	<pre>def greetPerson(*name): print('Hello', name) greetPerson('Frodo', 'Sauron')</pre> <p>Что будет выведено в терминале:</p>	<p>a. Hello Frodo Hello Sauron b. Hello ('Frodo', 'Sauron') c. Hello Frodo d. Syntax Error! greetPerson() can take only one argument.</p>
7.	<pre>result = lambda x: x * x print(result(5))</pre> <p>Что будет выведено в терминале?</p>	<p>a. lambda x: x*x b. 10 c. 25 d. 5*5</p>
8.	<pre>def unpack(a,b,c,d): print(a+d) x = [1,2,3,4] unpack(*x)</pre> <p>Что будет выведено в терминале?</p>	<p>a. Error b. [1,4] c. [5] d. 5</p>
9.	<pre>numbers = [1, 3, 6] newNumbers = tuple(map(lambda x: x , numbers)) print(newNumbers)</pre> <p>Что будет выведено в терминале?</p>	<p>a. [1, 3, 6] b. (1, 3, 6) c. [2, 6, 12] d. (2, 6, 12)</p>
10.	<pre>x = True y = False z = False if not x or y: print(1) elif not x or not y and z: print(2) elif not x or y or not y and x: print(3) else: print(4)</pre> <p>Что будет напечатано?</p>	<p>a. 1 b. 2 c. 3 d. 4</p>
11.	<pre>a = [1,2,3,None(),[],] print(len(a))</pre> <p>Что выведет следующая программа?</p>	<p>a. Syntax Error b. 4 c. 5 d. 6 e. 7</p>
12.	<pre>def a(b, c, d): pass</pre> <p>Что делает следующий код?</p>	<p>a. Определяет список и инициализирует его. b. Определяет функцию, которая ничего не делает. c. Определяет функцию,</p>



		которая передает параметры. d. Определяет пустой класс.
13.	Что будет напечатано? <pre>kvps = {"user", "bill", "password", "hillary"} print(kvps['password'])</pre>	a. user b. bill c. password d. hillary e. Ничего. TypeError.
14.	Что выведет следующий код, при его исполнении? Используется Python 3.x. <pre>print(type(1 / 2))</pre>	a. class 'int' b. class 'number' c. class 'float' d. class 'double' e. class 'tuple'
15.	Что будет напечатано? <pre>name = "snow storm" print("%s" % name[6:8])</pre>	a. st b. sto c. to d. Syntax Error
16.	Что выведет следующий фрагмент кода? <pre>x = 4.5 y = 2 print(x // y)</pre>	a. 2.0 b. 2.25 c. 9.0 d. 20.25 e. 21
17.	Что выведет следующий код? <pre>d = lambda p: p * 2 t = lambda p: p * 3 x = 2 x = d(x) x = t(x) x = d(x) print(x)</pre>	a. 7 b. 12 c. 24 d. 36 e. 48
18.	Какая из функций вернет итерируемый объект?	a. len() b. xrange() c. range() d. ord()
19.	Чему равна переменная L2 ? <pre>>>> L1 = [2, 3, 4] >>> L2 = L1 >>> L1[0] = 24 >>> L1 [24, 3, 4] >>> L2</pre>	a. [2, 3, 4] b. [24, 3, 4] c. [2, 3, 24] d. [2, 4, 3]
20.	Каким будет результат выполнения данной программы? <pre>list = [] for i in range(100): list.append(lambda x: x+i) list[42](3)</pre>	a. 45 b. 42 c. 102 d. Ошибка
21.	Чему равна переменная L2 ?	a. [2, 3, 4] b. [4, 3, 2]



	<pre>>>> L1 = [2, 3, 4] >>> L2 = L1[:] >>> L1[0] = 24 >>> L1 [24, 3, 4] >>> L2</pre>	<p>c. [24, 3, 4] d. [24, 4, 3]</p>
22.	<p>Что напечатает следующий код:</p> <pre>a = [1, 2, 3] a[2] = 2 print(a)</pre>	<p>a. [1, 2, 3] b. [1, 2, 2] c. [1, 2, 2, 3] d. Возникнет ошибка, так как списки являются immutable в Python 3</p>
23.	<p>Укажите модули входящие в стандартный набор Python:</p>	<p>a. math b. cmath c. numpy d. PIL</p>
24.	<p>Какие фрагменты кода создадут словари (dictionary)?</p>	<p>a. tel = {'qwe', 'asd', 'zxc'} b. tel = {'qwe': 1, 'asd': 2, 'zxc': 3} c. dict(['asd', 'qwe', 'zxc']) d. dict([('asd': 1), ('zxc': 2), ('qwe': 3)]) e. dict([('asd', 1), ('zxc', 2), ('qwe', 3)]) f. ни один из перечисленных</p>
25.	<p>Каким будет результат выполнения данной программы?</p> <pre>list = [] for i in range(100): list.append(lambda x, i = i : x+i) list[42](3)</pre>	<p>a. 45 b. 42 c. 102 d. Ошибка</p>
26.	<p>Что напечатает следующий код:</p> <pre>def dbl(self, x): return x * 2 class C: f = dbl obj = C() print(obj.f(1))</pre>	<p>a. 0 b. 2 c. ничего не будет напечатано d. возникнет ошибка, так как метод класса не может быть объявлен вне класса e. возникнет ошибка, так как метод f принимает 2 параметра</p>
27.	<p>Что выведет следующий код:</p> <pre>a = 3 a = "foo" if a / 2 == 1 else 2 a = a + a print(a)</pre>	<p>a. 6 b. Возникнет ошибка c. 2 d. 4 e. foofoo</p>
28.	<p>Что напечатает следующий код:</p>	<p>a. hello b. hell</p>



	<pre>text = 'hello' print(text[4:100])</pre>	<p>c. 0 d. возникнет ошибка e. foofoo</p>
29.	<p>Что напечатает следующий код:</p> <pre>for i in range(3): if i < 1: print(i) else: print(i) break</pre>	<p>a. 0 b. 0 1 c. 0 1 2 d. нечего не напечатает e. возникнет ошибка</p>
30.	<p>Что произойдет в результате выполнения следующего кода:</p> <pre>def summ(arg1, arg2): return arg1 + arg2 tup = 1, 2 print(summ(tup))</pre>	<p>a. будет напечатано 0 b. будет напечатано 3 c. нечего не будет напечатано d. возникнет ошибка</p>
31.	<p>Что напечатает следующий код:</p> <pre>7//3 + 7//-3</pre>	<p>a. 0 b. 1 c. -1 d. в Python3 нет оператора //</p>
32.	<p>Как можно вызвать метод func у следующего класса (выберите все подходящие варианты):</p> <pre>class myClass: def func(self): print('hello')</pre>	<p>a. myClass.func() b. obj = myClass(obj).func() c. obj = myClass(myClass.func(obj)) d. obj = myClass(obj).func e. ни один из перечисленных</p>
33.	<p>Каким будет результат выполнения кода:</p> <pre>a = [1, 2, 3] if a[2] < 3: print(a[a[1]]) else: print(a[1])</pre>	<p>a. 1 b. 2 c. 3 d. возникнет ошибка</p>
34.	<p>Что необходимо добавить на место пропущенной строки?</p> <pre>def find_max(nums): max_num = float("-inf") for num in nums: if num > max_num: # пропущенная строка return max_num</pre>	<p>a. max_num = num b. num = max_num c. max_num += 1 d. max_num += num</p>
35.	<p>Каким будет результат выполнения следующего кода:</p>	<p>a. 0 b. 1 c. func(0) d. возникнет ошибка</p>



	<pre>def func(n): n = n + 1 return n print(func(0))</pre>	
36.	Каким будет результат выполнения следующего кода: <pre>value = 0 value = value > 0 ? 0 : 1 print(value)</pre>	a. 0 b. 1 c. возникнет ошибка
37.	Что напечатает следующий код: <pre>fruits = {'apple', 'banana', 'apple'} print(fruits)</pre>	a. {'apple', 'banana', 'apple'} b. {'apple', 'apple', 'banana'} c. {'apple', 'banana'} d. возникнет синтаксическая ошибка
38.	Что напечатает следующий код: <pre>class myClass: i = 0 def __init__(self): i = 1 obj = myClass() print(obj.i)</pre>	a. 0 b. 1 c. None d. возникнет ошибка
39.	Укажите все правильные способы конкатенации двух строк	a. 'asd' + 'zxc' b. 'asd' . 'zxc' c. 'asd' 'zxc' d. 'asd'.strip() 'zxc' e. ничего из перечисленного
40.	Что напечатает следующий фрагмент кода: <pre>a = 0 try: print(a + 'a') except TypeError as err: print(err)</pre>	a. ничего не напечатает b. 0a c. будут напечатаны подробности возникшей ошибки d. в коде есть синтаксическая ошибка
41.	Каким будет результат: <pre>value = 0 def func(arg=1): arg = arg + 1 return arg print(func() + func(value))</pre>	a. 1 b. 2 c. 3 d. 4 e. Ошибка
42.	Укажите все правильные способы использования print:	a. print 5 b. print "5" c. print (5) d. print(5, "5") e. print() * 5
43.	Что напечатает следующий код:	a. 0 b. ничего не напечатает



	<pre>value = 0 if value > 0: print('>0') elif: print('<=0')</pre>	с. возникнет синтаксическая ошибка
44.	Что напечатает следующий код: <pre>a, b = 0, 1 while b < 5: print(b, end=' ') a, b = b, a+b</pre>	a. 1 2 3 4 b. 1 1 2 c. 1 1 2 3 d. возникнет ошибка
45.	Чему равна переменная b? <pre>>>> a = 3 >>> b = a >>> a = a + 2</pre>	a. 5 b. 3 c. 2 d. 0
46.	Каким будет результат выполнения кода: <pre>def func(arg1, arg2 = '1'): print(arg1, arg2) func(arg2 = '1', arg1 = '2')</pre>	b. 1 2 c. 2 1 d. возникнет ошибка
47.	Что напечатает следующий код: <pre>word = 'foobar' print(word[3:] + word[:3])</pre>	a. foobar b. barfoo c. fobar d. возникнет ошибка
48.	Что будет напечатано в результате выполнения следующего кода: <pre>1: qwe = 'qwe' 2: qwe[0] = 'a' 3: print(qwe)</pre>	a. qwe b. awe c. возникнет ошибка в строке 2 d. возникнет ошибка в строке 3
49.	Каким будет результат выполнения следующего фрагмента кода: <pre>for i in range(3): print(i, end=' ')</pre>	a. 3 b. 0 1 2 c. 1 2 3 d. 0 1 2 3 e. возникнет ошибка
50.	Какое значение получит a? a = 2,3	a. 2 b. 3 c. (2,3)
51.	Блокнот Jupyter Notebook можно сохранить в формате:	a. IPython Notebook (.ipynb) b. html c. текстовый файл d. проект PyCharm e. проект Visual Studio
52.	Как правильно делать проверку на POST-запрос в представлении (Django)?	a. if request.POST: b. if "POST" in request: c. if "POST" in



		<code>request.method:</code> d. if request.method == "POST":
53.	Для того, чтобы текст передаваемый в шаблон воспринимался как HTML следует(Django):	a. использовать функцию <code>to_html()</code> b. "обратить место в шаблоне, где выводится текст тегом <code>{% autoescape off %}</code> ... <code>{% endautoescape %}</code> c. использовать фильтр <code>unescape</code> d. Добавить в начало шаблона тег <code>{% autoescape off %}</code>
54.	Какая из следующих команд запускает сервер разработки, разрешая внешние соединения(Django)?	a. <code>manage.py runserver 8000</code> b. <code>manage.py runserver 0.0.0.0:8000</code> c. <code>manage.py runserver 8000 -nonlocal</code> d. <code>manage.py runserver 8000 -external</code>
55.	Как сделать поле модели необязательным(Django)?	a. Интерфейс администратора рассматривает все поля как необязательные b. Добавить к полю <code>blank=True</code> c. Добавить к полю <code>req=False</code> d. Добавить к полю <code>required=False</code>



4. Порядок проведения и критерии оценивания промежуточной аттестации

4.1. Порядок проведения промежуточной аттестации

Зачёт проводится в виде тестирования. Студент должен ответить на вопросы закрытого типа, которые предполагают выбор вариантов ответа, а также на вопросы открытого типа, которые не предполагают вариантов ответа, правильный ответ требуется написать самостоятельно. Всего 20 тестовых вопросов. Продолжительность теста – 35 минут.

4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств

Тест формируется в системе электронного обучения MOODLE.

Максимальный балл за тест — 100 баллов.

Оценка	Зачтено	Незачтено
Баллы	100-60 баллов	59-0 баллов
Уровень освоения проверяемых компетенций	высокий	низкий

4.3. Результаты промежуточной аттестации и уровни сформированности компетенций

При подведении итогов учитываются результаты только промежуточной аттестации:

0-59 баллов – незачет;

60-100 баллов – зачет;

Особенности проведения процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья обозначены в рабочей программе дисциплины (модуля).

Уровни сформированности компетенций определяется следующим образом:

1. Высокий уровень сформированности компетенций соответствует оценке зачтено:
 - предполагает формирование компетенций на высоком уровне;
 - знание теоретических разделов изучаемой дисциплины на уровне не ниже оценки удовлетворительно;
 - студент умеет применять на практике знания, полученные в рамках изучения дисциплины
 - формируются навыки использования теоретических и практических разделов дисциплины для решения задач профессиональной деятельности;
2. Низкий уровень соответствует оценке незачтено.