

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

УТВЕРЖДАЮ  
Проректор по учебной работе и  
качеству образования

\_\_\_\_\_ И. А. Долгова

16 апреля 2025 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЫ

**ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ**

---

Направление подготовки:	09.03.03 Прикладная информатика
Профиль подготовки:	Корпоративные информационные системы
Квалификация:	бакалавр
Форма обучения:	очная, очно-заочная
Год начала подготовки:	2025

Самара  
2025

# 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
ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.И-1. Разрабатывает алгоритмы для компьютерных программ	ОПК-7.И-1.3-1. Знает основы алгоритмизации информационно-вычислительных задач	Текущий контроль: промежуточный тест, лабораторная работа. Промежуточная аттестация: зачет
		ОПК-7.И1.У-1. Умеет разработать оптимальный алгоритм для решения профессиональной информационно-вычислительной задачи	
	ОПК-7.И-2. Разрабатывает программы, пригодные для практического применения	ОПК-7.И-2.3-1. Знает основы программирования на алгоритмическом языке высокого уровня	Текущий контроль: промежуточный тест, лабораторная работа. Промежуточная аттестация: зачет
		ОПК-7.И-2.У-1. Умеет создавать компьютерные программы для решения практических задач	

## 2. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ

### 2.1. Темы лабораторных работ

#### Раздел 1. Введение в объектно-ориентированное программирование (ООП).

Лабораторная работа №1. Введение в ООП на языке Python.

*Цель работы:* Ознакомиться с объектно-ориентированной парадигмой программирования.

*Вопросы для самопроверки:*

1. В чем отличие объектно-ориентированной парадигмы программирования от процедурной?
2. В чем преимущества объектно-ориентированной парадигмы?
3. Как объявляются классы в Python?
4. Что такое конструктор класса?
5. Какие основные правила реализации конструктора?

#### Раздел 2. Атрибуты класса и объекта.

Лабораторная работа №2. Классы и объекты

*Цель работы:* Научиться работать с атрибутами классов и объектов.

*Вопросы для самопроверки:*

1. Что такое атрибут объекта?
2. Чем атрибут объекта отличается от атрибута класса?
3. Можно ли атрибут класса изменить через объект?
4. Можно ли атрибут объекта изменить через класс?
5. Что такое метод класса?

#### Раздел 3. Наследование.

Лабораторная работа №3. Базовый принцип ООП - наследование.

*Цель работы:* Научиться пользоваться принципом наследования при разработке программ.

*Вопросы для самопроверки:*

1. Что такое класс-потомок?
2. Что такое класс-предок?
3. Как создать класс потомок на основе класса предка?
4. Что такое функция super?
5. Как переопределить методы класса-потомка?

#### Раздел 4. Инкапсуляция.

Лабораторная работа №4. Базовый принцип ООП - инкапсуляция.

*Цель работы:* Научиться пользоваться принципом инкапсуляции при разработке программ.

*Вопросы для самопроверки:*

1. В чем суть принципа инкапсуляции?
2. Как создать внутренние (приватные) атрибуты объекта?
3. Чем отличаются атрибуты с одинарным и двойным символом подчеркивания?
4. Что такое геттер?
5. Что такое сеттер?

#### Раздел 5. Полиморфизм.

Лабораторная работа №5. Базовый принцип ООП - полиморфизм.

*Цель работы:* Научиться пользоваться принципом полиморфизма при разработке программ.

*Вопросы для самопроверки:*

1. В чем суть принципа полиморфизма в Python?
2. Как можно реализовать полиморфную функцию?
3. Что такое "утиная типизация"?
4. Что такое магические (dunder) методы в Python?
5. Как работает магический метод `__add__`?

#### Раздел 6. Введение в базы данных.

Лабораторная работа №6. Работа с базами данных на Python

*Цель работы:* Научиться использовать базы данных в программах.

*Вопросы для самопроверки:*

1. В чем основные преимущества базы данных SQLite?
2. Как программно создать базу данных?
3. Как реализовать запрос к базе данных?
4. Как объединить таблицы в базе данных?
5. Как удалить ряд записей в базе данных?

#### Раздел 7. Технология ORM.

Лабораторная работа №7. Работа с базами данных с использованием ORM

*Цель работы:* Научиться технике работы с базами данных по методологии ORM.

*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 8. Графический интерфейс Tkinter.

Лабораторная работа №8. Работа с библиотекой Tkinter.

*Цель работы:* Научится создавать программы с графическим интерфейсом.

*Вопросы для самопроверки:*

1. Зачем нужен графический интерфейс при программировании?
2. Какие графические библиотеки поддерживаются Python?
3. Как работает упаковщик `pack()`?
4. Как работает упаковщик `grid()`?
5. Как работает упаковщик `place()`?

#### Раздел 9. Дополнительные виджеты Tkinter.

Лабораторная работа №9. Продолжение работы с библиотекой Tkinter.

*Цель работы:* Научится использовать дополнительные виджеты библиотеки Tkinter.

*Вопросы для самопроверки:*

1. Как работают виджеты Scale, Spinbox?
2. Как реализовать строку меню?
3. Как вызвать окна открытия и закрытия файлов?
4. Что такое модальные окна?
5. Что такое фреймы?

### 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**

<b>Критерии</b>	<b>Баллы</b>
Степень соответствия выполненного задания поставленным требованиям	25
Структурирование и комментирование лабораторной работы	25

Критерии	Баллы
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.(ОПК-7.1)

Установите соответствие между понятиями традиционной (процедурной) и объектно-ориентированной парадигмами программирования:

1. Метод	А. Процедурная
2. Функция	
3. Объект	Б. Объектно-ориентированная
4. Подпрограмма	
5. Класс	

А	Б

##### 2. Прочитайте текст и установите соответствие. (ОПК-7.1)

Установите соответствие между понятиями (1-6) и утверждениями (А,Б):

1. Класс	А. Относятся к базовым понятиям ООП
2. Объект	
3. Рекурсия	Б. Не относятся к базовым понятиям ООП
4. Наследование	
5. Инкапсуляция	
6.Полиморфизм	

А	Б

##### 3. Прочитайте текст и установите соответствие. (ОПК-7.1)

Установите соответствие между утверждениями (1-7) и понятиями (А,Б):

1. Объект - это более узкое понятие, чем класс	А. Истина (True)
2. Конструктор – это переменная класса	
3. Класс служит для установки начала выполнения программы	Б. Ложь (False)
4. У объекта может быть только один конструктор	
5. В программе может существовать большое количество объектов одного класса	
6.Объект класса создается с помощью конструктора	
7. В Java конструктор использует ключевое слово self	

А	Б

##### 4. Прочитайте текст и установите последовательность (ОПК-7.1)

Установите правильную последовательность при разработке объектно-ориентированного приложения

А. Разработка объектов базового класса

- Б. Разработка алгоритма приложения  
 В. Разработка алгоритма взаимодействия между объектами  
 Г. Разработка методов и атрибутов базового класса  
 Д. Разработка наследуемых классов и объектов на их основе

--

### 5. Прочитайте текст и установите соответствие. (ОПК-7.1)

Установите соответствие между ключевыми словами (1-7) и языками программирования (А,Б):

1. self	А. Язык объектно-ориентированного программирования Java
2. this	
3. public	
4. private	
5. protected	
6.global	
7. __init__	
	Б. Язык объектно-ориентированного программирования Python

А	Б

### 6. Прочитайте текст и установите соответствие. (ОПК-7.1)

Установите соответствие между фрагментами кода (1,2,3,4,5) и алгоритмами (А,Б,В,Г,Д):

1. <pre>{ int m=a[0];   for(int i=1; i&lt;10; i++)     if(a[i]&gt;m) m=a[i]; }</pre>	А. Фрагмент реализации алгоритма нахождения суммы элементов массива (Java)
2. <pre>ix=0 mel=slm[0] for i,el in enumerate(slm):   if el&lt;mel:     ix=i</pre>	Б. Фрагмент реализации алгоритма нахождения произведения элементов списка ( Python)
3. <pre>{ int s=0;   for(int i=0; i&lt;10; i++)     s+=a[i]; }</pre>	В. Фрагмент реализации алгоритма нахождения количества отрицательных элементов списка ( Python)
4. <pre>k=1 for el in slm:   k*=el</pre>	Г. Фрагмент реализации алгоритма нахождения максимального элемента массива (Java)
5. <pre>n=0 for el in slm:   if el&lt;0:     n+=1</pre>	Д. Фрагмент реализации алгоритма нахождения индекса минимального элемента списка (Python)

А	Б	В	Г

### 7. Прочитайте текст и установите последовательность (ОПК-7.1)



Установите правильную последовательность запуска приложения с графическим интерфейсом tkinter на языке программирования Python

- А. win=tkinter.Tk()
- Б. win.mainloop()
- В. import tkinter
- Г. win.geometry(f '{w}x{h}')
- Д. w=800; h=400

--

**8. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ОПК-7.1)**

Для уменьшения объема кода объектно-ориентированного приложения используется такое понятие как:

- А. Полиморфизм
- Б. Атрибут
- В. Инкапсуляция
- Г. Конструктор
- Д. Наследование


**9. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ОПК-7.2)**

Для реализации внутренних и внешних (интерфейсных) методов и атрибутов используется такое понятие как:

- А. Полиморфизм
- Б. Наследование
- В. Инкапсуляция
- Г. Конструктор
- Д. Атрибут


**10. Прочитайте текст, выберите правильный ответ и запишите аргументы, обосновывающие выбор ответа. (ОПК-7.2)**

Для реализации принципа динамического связывания в программировании используется такое понятие как:

- А. Полиморфизм
- Б. Наследование
- В. Инкапсуляция
- Г. Конструктор
- Д. Атрибут


**11. Прочитайте текст и запишите развёрнутый обоснованный ответ. (ОПК-7.2)**

При практической реализации десктопного приложения на языке Python часто применяется модуль tkinter. Зачем он используется и чем обусловлено его широкое распространение?

--

**12. Прочитайте текст и запишите развёрнутый обоснованный ответ. (ОПК-7.2)**

При практической реализации приложений на языке Python часто используется модуль sqlite3. Зачем он используется и чем обусловлено его широкое распространение?

--

**13. Прочитайте текст и установите соответствие. (ОПК-7.2)**

Установите соответствие между языками программирования (1-2) и вопросом, иллюстрирующим реализацию принципа полиморфизма (А-Б):

1. Java	А. "Кто ты?"
2. Python	Б. "Что умеешь делать?"

А	Б

**14. Прочитайте текст и запишите развёрнутый обоснованный ответ. (ОПК-7.2)**

Что, по Вашему, делает следующий фрагмент программного кода:

```
students= (Student
            .select()
            .join(Oценка)
            .order_by(Student.name)
            .where((Student.mark== Oценка.mInt) & (Student.mark >=4)))
for student in students
print(f'{student.fam}: {student.mark.mStr} ")
```

--

**15. Установите соответствие между функциями (1-2) и их характеристикой (А-Б): (ОПК-7.2)**

1. <code>__init__(self)</code>	А. Относится к dunder - методам
2. <code>__spot__(*arg)</code>	Б. Не относится к dunder - методам

А	Б
---	---

### 3.2. Ключи к контрольным заданиям

№ задания	Верный ответ
1	A24B135
2	A12456B3
3	A156B2347
4	БАГДВ
5	A2345B167
6	A3B4B5Г1Д2
7	ВАДГБ
8	Д Принцип наследования в объектно-ориентированном программировании позволяет создавать новые классы на основе уже существующих (ранее разработанных). Таким образом, код не дублируется, что уменьшает его объем
9	В Инкапсуляция позволяет разделить методы и атрибуты класса на внешние (интерфейсные) и внутренние. Внешние методы и атрибуты позволяют взаимодействовать с другими классами, а внутренние – обеспечивают скрытую реализацию алгоритма. Таким образом проводится упорядочивание компонент и повышается надежность работы приложения
10	А Принцип полиморфизма, в основе которого лежит механизм динамического связывания позволяет разработчику использовать методы с одним и тем же названием, которые применяются к разным объектам. В ходе выполнения кода метод сам "понимает" с каким объектом имеет дело и ведет себя соответствующим образом. Это существенно повышает эффективность разработки объектно-ориентированных приложений
11	tkinter – это модуль языка Python, предназначенный для разработки графического интерфейса приложения. По сравнению с другими модулями, такими как WxPython, PyQt и другими, tkinter обладает следующими достоинствами: а) Встроен в библиотеку Python б) Прост в реализации в) Обладает набором всех необходимых виджетов г) Интегрирован непосредственно в программный код
12	sqlite3 – это модуль языка Python, предназначенный для разработки приложений с использованием баз данных. По сравнению с другими базами данных, такими как MySQL, PostgreSQL и другими, SQLite обладает следующими достоинствами: а) "Легкая" (занимает мало места) б) Реляционная в) Файловая г) Поддерживает визуальные конструкторы
13	A1B2
14	Формируется запрос <b>students</b> 1. Выбираются все записи из таблицы <b>Student</b> 2. Они связываются с таблицей <b>Ocenka</b> 3. Записи сортируются по полю name таблицы <b>Student</b> 4. Отбираются только те записи, для которых оценка больше или равно 4 5. Записи последовательно выводятся в консоль по 2 полям: имя

	студента и текстовое описание его оценки
15	A1B2

### 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 зачету)

1. Процедурная и объектно-ориентированные парадигмы программирования.
2. Преимущества ООП.
3. Классы в Python.
4. Характеристики класса – атрибуты и методы.
5. Конструктор класса.
6. Правила реализации конструктора.
7. Специальные атрибуты класса.
8. Атрибуты класса и их отличие от атрибутов объекта.
9. Доступность атрибутов класса через объект.
10. Изменяемые и неизменяемые атрибуты.
11. Примеры программ с использованием ООП.
12. Методы класса, декоратор `@classmethod`.
13. Статические методы класса – декоратор `@staticmethod`.
14. Базовый принцип ООП – наследование.
15. Формирование класса-потомка, использование конструктора класса-предка, ключевое слово `super`.
16. Базовый принцип наследования – IS-A.
17. Использование класса-наследника в качестве класса-предка.
18. Множественное наследование.
19. Базовый принцип ООП – инкапсуляция.
20. Разделение методов объекта на внешние (публичные) и внутренние (приватные).
21. Использование знака подчеркивания ( `_` ) для обозначения внутренних атрибутов. Различие одинарного и двойного подчеркивания при формировании внутренних атрибутов.
22. Доступ к внутреннему атрибуту через имя класса.
23. Приватные атрибуты при наследовании.
24. Понятие о геттерах и сеттерах.
25. Приватные методы.
26. Использование модуля `accessify` для реализации приватных атрибутов.
27. Работа декоратора `@private`.
28. Работа с приватными атрибутами через объект `property`.
29. Понятие о дескрипторах.
30. Базовый принцип ООП – полиморфизм.
31. Особенности реализации принципа в Python, отличие от других языков программирования.
32. Полиморфная функция.
33. "Утиная" типизация (`ducktyping`).
34. Реализация абстрактного метода в Python.
35. Принципы использования магических (dunder) методов в Python.
36. Примеры реализации магических методов: `__add__()`, `__eq__()`, `__getitem__()`, `__iter__()`, `__next__()`.
37. Общие принципы использования баз данных в Python (на примере SQLite).
38. Методы `connect()`, `cursor()`, `execute()`, `commit()`, `close()`.
39. Методы вывода данных из запроса.
40. Формирование запроса с параметрами.
41. Множественная загрузка.
42. Объединение, сортировка, фильтрация, удаление данных.
43. Введение в метаклассы.
44. Пример динамического создания класса.
45. Вложенные классы.

46. Принципы использования технологии ORM (Object-relational mapper) – на примере модуля реееее.
47. Преимущества ORM – библиотек.
48. Создание базы данных и ее таблиц по технологии ORM.
49. Основные операции с базой данных (CRUD).
50. Обзор основных GUI-фреймворков Python.
51. Достоинства и недостатки Tkinter.
52. Создание графического окна и управление его параметрами.
53. Виджеты и способы их упаковки.
54. Особенности упаковщиков pack(), grid(), place().
55. Использование фреймов при упаковке.
56. Примеры на использование базовых виджетов: Label, Button, Entry, Radiobutton, Checkbutton, Combobox.
57. Примеры на использование дополнительных виджетов Tkinter: Scale, Spinbox.
58. Организация меню в Tkinter.
59. Окна открытия и закрытия файлов.
60. Фреймы. Модуль ttk.
61. Модальные окна.
62. Работа с виджетом Treeview.
63. Пример разработки программы с графическим интерфейсом.