

**Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сахалинский государственный университет»
Кафедра информатики**

УТВЕРЖДЕН
на заседании кафедры
«22» мая 2025 г., протокол № 9
Исполняющий обязанности
заведующего кафедрой информатики



Осипов Г.С.

**ФОНД
ОЦЕНОЧНЫХ СРЕДСТВ
ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

Б1.В.ДВ.03.01 Программирование на языке Python

Направление подготовки
09.03.03 Прикладная информатика

профиль
Автоматизированные системы обработки информации и управления

**Уровень высшего образования
БАКАЛАВРИАТ**

Южно-Сахалинск, 2025 г.

1. Формируемые компетенции и индикаторы их достижения по дисциплине

Код компетенции	Содержание компетенции	Код и наименование индикатора достижения компетенции
ПКС-2	Способен проводить формализацию предметной области с целью создания информационной системы	ПКС-2.1 - Знает требования к компьютерному программному обеспечению; виды технической спецификации на программные компоненты и их взаимодействие; методы проектирование компьютерного программного обеспечения ПКС-2.2 – Умеет применять требования к компьютерному программному обеспечению; разрабатывать технические спецификации на программные компоненты и их взаимодействие; применять методы проектирования компьютерного программного обеспечения; ПКС-2.3 – Владеет методами разработки требований к компьютерному программному обеспечению, технических спецификаций на программные компоненты, методами проектирования компьютерного программного обеспечения.

2. Паспорт фонда оценочных средств по дисциплине (модулю)

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1.	Тема 1. Язык Python. Типы данных, операции, операторы. Особенности ввода/вывода	ПКС-2	Задания к лабораторным работам, контрольные вопросы, вопросы к зачету
2.	Тема 2. Встроенные типы объектов: числа, строки, кортежи, списки, словари, множества.	ПКС-2	Задания к лабораторным работам, контрольные вопросы, вопросы к зачету
3.	Тема 3. Файловый ввод-вывод. Чтение строк с помощью итераторов файлов. Работа с двоичными файлами.	ПКС-2	Задания к лабораторным работам, контрольные вопросы, вопросы к зачету
4.	Тема 4. Библиотека numpy для реализации математических объектов и вычислений.	ПКС-2	Задания к лабораторным работам, контрольные вопросы, вопросы к зачету
5.	Тема 5. Создание приложений с GUI. Обзор графических библиотек: Tkinter, PyQt.	ПКС-2	Задания к лабораторным работам, контрольные вопросы, вопросы к зачету
6.	Тема 6. Классы в Python. Определение данных, методов, операций. Наследование. Множественное наследование. Композиция при разработке классов	ПКС-2	Задания к лабораторным работам, контрольные вопросы, вопросы к зачету
7.	Итоговая работа	ПКС-2	Задания к лабораторным работам, контрольные вопросы, вопросы к зачету
8.	зачет	ПКС-2	Задания к лабораторным работам, контрольные вопросы, вопросы к зачету

1. Оценочные средства

Тестирование является одним из средств формального контроля качества обучения – методе, основанном на стандартизированных заданиях, которые позволяют оценить знания, умения и навыки испытуемого

Требования к составлению тестового задания

Тесты должны быть построены по методике, обеспечивающей выполнение требований соответствующего федерального государственного образовательного стандарта.

В тестовых заданиях используются четыре типа вопросов:

– **закрытая** форма – является наиболее распространенной и предлагает несколько альтернативных ответов на поставленный вопрос. Тестовое задание, содержащее вопрос в закрытой форме, включает в себя один или несколько правильных ответов.

Закрытая форма вопросов используется также в тестах-задачах с выборочными ответами. В тестовом задании в этом случае сформулированы условие задачи и все необходимые исходные данные, а в ответах представлены несколько вариантов результата решения в числовом или буквенном виде.

Обучающийся должен решить задачу и показать, какой из представленных ответов он получил.

– **открытая** форма – вопрос представляет собой утверждение, которое необходимо дополнить. Данная форма может быть представлена в тестовом задании, например, в виде словесного текста, формулы (уравнения), графика, в которых пропущены существенные составляющие – части слова или буквы, условные обозначения, линии или изображения элементов схемы или графика.

Обучающийся должен по памяти вставить соответствующие элементы в указанные места («пропуски»).

– **установление соответствия** – в данном случае обучающемуся предлагают два списка, между элементами которых следует установить соответствие;

– **установление последовательности** – предполагает необходимость установить правильную последовательность предлагаемого списка слов или фраз.

Примерный тест

Задания для проверки сформированности компетенции ПКС-2 Способен проводить формализацию предметной области с целью создания информационной системы

Базовые концепции (Темы 1-2)

1. Выберите правильный ответ:

Укажите оператор для целочисленного деления

1. /
2. //
3. %
4. div

Ответ: 2.

2. Выберите правильный ответ:

Укажите ответ, который получится после выполнения фрагмента кода:

```
s = "Python"
print(s[1:4])
```

1. Pyt
2. yth
3. ytho
4. thon

Ответ: 2.

3. Выберите правильный ответ:

Определите неизменяемый (immutable) тип данных

1. Список (list)
2. Словарь (dict)
3. Множество (set)
4. Кортеж (tuple)

Ответ: 4.

4. Запишите ответ:

Запишите результат выполнения кода: _____

```
a = [1, 2, 3]
b = a
b.append(4)
print(a)
```

Ответ: [1, 2, 3, 4] (т.к. изменяемые объекты передаются по ссылке)

5. Запишите ответ:

Запишите результат выполнения кода: _____

```
d = {'a': 1, 'b': 2}
d['c'] = 3
print(d.get('d', 'Not found'))
```

Ответ: 'Not found' (метод get возвращает значение по умолчанию)

6. Запишите ответ:

Напишите функцию, которая принимает список чисел и возвращает новый список, содержащий только четные числа из исходного.

```
def get_even_numbers(numbers):
    # Ваш код здесь
```

Ответ: Возможный вариант ответа:

```
def get_even_numbers(numbers):
    return [x for x in numbers if x % 2 == 0]
```

Работа с файлами (Тема 3)

7. Выберите правильный ответ:

Определите режим открытия файла, который используется для чтения и записи бинарных данных

1. 'r'
2. 'w+'
3. 'rb+'
4. 'a'

Ответ: 3

8. Выберите правильный ответ:

Определите метод, который используется для чтения всех строк файла в список:

1. read()
2. readline()
3. readlines()
4. read_all()

Ответ: 3

Практическое задание:

9. Запишите ответ:

Напишите код, который:

- Открывает файл data.txt для чтения
- Считывает все строки
- Выводит только те строки, которые начинаются с символа '#'
- Закрывает файл

Сделайте вариант с использованием менеджера контекста (with)

Ваш код:

Ответ: Результат может быть таким:

```
with open('data.txt', 'r') as f:
    for line in f:
        if line.startswith('#'):
            print(line.strip())
```

Часть 3: NumPy (Тема 4)

10. Выберите правильный ответ:

Выберете результат выполнения кода:

```
import numpy as np
arr = np.zeros((3, 4))
print(arr.shape)
```

1. (3,)
2. (4,)
3. (3, 4)
4. (12,)

Ответ: 3.

11. Выберите правильный ответ:

Выберете вариант получения всех элементов массива arr больше 5:

1. arr > 5
2. arr[arr > 5]
3. arr.where(arr > 5)
4. filter(arr > 5)

Ответ: 2

Практическое задание:

12. Запишите ответ:

Создайте numpy-массив размером 5x5, в котором:

- На главной диагонали стоят 1
- Над диагональю стоят 2
- Под диагональю стоят 3

Пример для 3x3:

```
[[1, 2, 2],  
 [3, 1, 2],  
 [3, 3, 1]]
```

Ваш код:

Ответ: Результат может быть таким:

```
import numpy as np  
n = 5  
arr = np.zeros((n, n))  
np.fill_diagonal(arr, 1)  
for i in range(n):  
    for j in range(n):  
        if j > i:  
            arr[i, j] = 2  
        elif j < i:  
            arr[i, j] = 3
```

Часть 4: GUI (Тема 5)

13. Выберите правильный ответ:

Укажите виджет Tkinter, который используется для ввода одной строки текста

1. Label
2. Button
3. Entry
4. Text

Ответ: 3

14. Выберите правильный ответ:

Укажите метод упаковки виджетов в Tkinter?

1. pack(), grid(), или place()
2. show()
3. display()
4. render()

Ответ: 1

Часть 5: ООП (Тема 6)

15. Выберите правильный ответ:

Укажите метод, который является конструктором класса в Python

1. `__init__`
2. `__new__`
3. `__create__`
4. `__constructor__`

Ответ: 1.

16. Выберите правильный ответ:

Определите понятие множественного наследования:

1. Наследование от нескольких классов одновременно
2. Создание нескольких экземпляров одного класса
3. Наследование методов от встроенных типов
4. Использование нескольких конструкторов

Ответ: 1

17. Запишите ответ:

Определите результат выполнения кода:

```
class A:
    def __init__(self):
        print("A")

class B(A):
    def __init__(self):
        super().__init__()
        print("B")

b = B()
```

Ответ:

A

B

Проектирование классов:

18. Запишите ответ:

Спроектируйте систему классов для библиотеки:

- Класс Book (атрибуты: title, author, year)
- Класс Library (содержит список книг, методы: add_book, find_book_by_author)
- - Реализуйте метод `__str__` для Book
- - Используйте композицию (Library содержит Book)

Ваш код:

Ответ: Результат может быть таким:

Класс книги

```
class Book:
    def __init__(self, title, author, year):
```

```

        self.title = title
        self.author = author
        self.year = year

    def __str__(self):
        return f'"{self.title}" - {self.author} ({self.year})'

# Класс библиотеки (композиция: библиотека содержит книги)
class Library:
    def __init__(self):
        self.books = [] # Список объектов Book

    def add_book(self, book):
        self.books.append(book)

    def find_book_by_author(self, author):
        return [book for book in self.books if book.author == author]

# Пример использования
if __name__ == "__main__":
    # Создаем книги
    book1 = Book("Война и мир", "Лев Толстой", 1869)
    book2 = Book("Преступление и наказание", "Фёдор Достоевский", 1866)
    book3 = Book("Анна Каренина", "Лев Толстой", 1877)

    # Создаем библиотеку
    library = Library()

    # Добавляем книги
    library.add_book(book1)
    library.add_book(book2)
    library.add_book(book3)

    # Выводим книги
    print("Все книги в библиотеке:")
    for book in library.books:
        print(f" {book}")

    # Ищем книги по автору
    print("\nКниги Толстого:")
    for book in library.find_book_by_author("Лев Толстой"):
        print(f" {book}")

```

19. Запишите ответ:

Создайте класс Vehicle и два дочерних класса Car и Bicycle:

- Vehicle имеет атрибуты brand, model и метод get_info()
- Car добавляет атрибут num_doors

- Bicycle добавляет атрибут num_gears
- - Переопределите get_info() в дочерних классах

Ваш код:

Ответ: Результат может быть таким:

```
# Базовый класс
class Vehicle:
    def __init__(self, brand, model):
        self.brand = brand
        self.model = model

    def get_info(self):
        return f"Марка: {self.brand}, Модель: {self.model}"

# Класс автомобиля
class Car(Vehicle):
    def __init__(self, brand, model, num_doors):
        super().__init__(brand, model)
        self.num_doors = num_doors

    def get_info(self):
        base_info = super().get_info()
        return f"{base_info}, Количество дверей: {self.num_doors}"

# Класс велосипеда
class Bicycle(Vehicle):
    def __init__(self, brand, model, num_gears):
        super().__init__(brand, model)
        self.num_gears = num_gears

    def get_info(self):
        base_info = super().get_info()
        return f"{base_info}, Количество передач: {self.num_gears}"

# Пример использования
if __name__ == "__main__":
    # Создание объектов
    car = Car("Toyota", "Camry", 4)
    bicycle = Bicycle("Giant", "Escape", 21)

    # Вывод информации
    print(car.get_info()) # Марка: Toyota, Модель: Camry, Количество дверей:
4
    print(bicycle.get_info()) # Марка: Giant, Модель: Escape, Количество
передач: 21

    # Демонстрация полиморфизма
    vehicles = [car, bicycle]
    for vehicle in vehicles:
        print(f"Транспортное средство: {vehicle.get_info()}")
```

Критерии оценки знаний при проведении тестирования

Отметка «отлично» выставляется при условии правильного ответа не менее чем 85% тестовых заданий;

Отметка «хорошо» выставляется при условии правильного ответа не менее чем 70 % тестовых заданий;

Отметка «удовлетворительно» выставляется при условии правильного ответа не менее 50 %;

Отметка «неудовлетворительно» выставляется при условии правильного ответа менее чем на 50 % тестовых заданий.

Результаты текущего контроля используются при проведении промежуточной аттестации