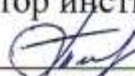


МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ ВЛАДИМИРА ДАЛЯ»

Институт приборостроения и электротехнических систем
Кафедра «Приборы»

УТВЕРЖДАЮ
Директор института


(подпись)



« 25 » февраля 20 25 года

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по учебной дисциплине

«Математическое моделирование сигналов и систем»
12.04.01 Приборостроение
«Инновационные технологии в приборостроении»,
«Измерительные информационные технологии»

Разработчики:
проф.  Петров А.С.
(подпись)

ФОС рассмотрен и одобрен на заседании кафедры «Приборы»
от « 25 » февраля 20 25 г., протокол № 6

Заведующий кафедрой  Eroshin S.S.
(подпись)

Луганск 2025 г.

**Комплект оценочных материалов по дисциплине
«Математическое моделирование сигналов и систем»**

Задания закрытого типа

Задания закрытого типа на выбор правильного ответа

Выберите один правильный ответ

1. Какая модель является предметом формализации?

- А) описательная
- Б) математическая
- В) графическая

Правильный ответ: Б

Компетенции (индикаторы): ОПК-3, ПК-3

2. Выходным слоем обобщённого многослойного персептрона называется:

- А) первый слой сети
- Б) все слои сети
- В) последний слой сети

Правильный ответ: В

Компетенции (индикаторы): ОПК-3, ПК-3

3. Какие системы искусственного интеллекта (СИИ) входят в состав систем эвристического поиска?

- А) нейросистемы
- Б) игровые системы
- В) системы распознавания
- Г) экспертные системы

Правильный ответ: А, Г

Компетенции (индикаторы): ОПК-3, ПК-3

4. Натурное (материальное) моделирование:

- А) моделирование, при котором в модели узнается какой-либо отдельный признак объекта-оригинала
- Б) моделирование, при котором в модели узнается моделируемый объект, то есть натурная (материальная) модель всегда имеет визуальную схожесть с объектом-оригиналом
- В) создание математических формул, описывающих форму или поведение объекта-оригинала

Правильный ответ: Б

Компетенции (индикаторы): ОПК-3, ПК-3

5. Система состоит из:

А) объектов, которые называются свойствами системы

Б) набора отдельных элементов

В) объектов, которые называются элементами системы

Правильный ответ: В

Компетенции (индикаторы): ОПК-3, ПК-3

6. Научное направление, связанное с попытками формализовать мышление человека называется:

А) представлением знаний

Б) нейронной сетью

В) экспертной системой

Г) искусственным интеллектом

Правильный ответ: Г

Компетенции (индикаторы): ОПК-3, ПК-3

7. Какие модели воспроизводят геометрические, физические и другие свойства объектов в материальной форме?

А) табличные

Б) предметные

В) информационные

Правильный ответ: Б

Компетенции (индикаторы): ОПК-3, ПК-3

8. Что такое нечёткое множество?

А) Множество значений, определяемых случайными величинами

Б) Совокупность словесных характеристик в виде базовой шкалы, и функция принадлежности их данной шкале

В) Множество значений, определяемых временными соотношениями

Г) Совокупность словесных характеристик из заданного алфавита и числовыми характеристиками

Правильный ответ: Б

Компетенции (индикаторы): ОПК-3, ПК-3

Выберите все правильные варианты ответов

9. Назовите характеристики систем массового обслуживания (СМО):

А) Одноканальные и многоканальные СМО с отказами.

Б) Одноканальные и многоканальные СМО с неограниченной очередью.

В) Одноканальные и многоканальные СМО с ограниченной очередью.

Г) Одноканальная СМО замкнутого типа с m источниками заявок.

Правильные ответы: А, Б, В, Г

Компетенции (индикаторы): ОПК-3, ПК-3

Задания закрытого типа на установление соответствия

Установите правильное соответствие.

Каждому элементу левого столбца соответствует только один элемент правого столбца.

1. Установите соответствие применения методов Монте-Карло, метода Рунге-Кутты, метода сеток, метода генетических алгоритмов.

1) Этот метод используется для численной оценки интегралов, решения задач статистической физики, моделирования случайных процессов и оптимизации. Он особенно полезен в ситуациях, когда аналитическое решение сложно или невозможно получить.

А) Метод генетических алгоритмов

2) Этот метод используется для численного решения обыкновенных дифференциальных уравнений (ОДУ). Он позволяет находить приближенные решения уравнений, которые описывают динамику различных физических и инженерных систем.

Б) Метод конечных разностей

3) Используется для решения дифференциальных уравнений и задач, связанных с частными производными. Метод подразумевает разбиение области на сетку и аппроксимацию значений функции в узлах сетки.

В) Метод Рунге-Кутты

4) Этот метод относится к классу эволюционных алгоритмов и используется для оптимизации и поиска решений в сложных многомерных пространствах. Он имитирует процесс естественного отбора, чтобы находить оптимальные решения.

Г) Метод Монте-Карло

Правильный ответ: 1-Г, 2-В, 3-Б, 4-А

Компетенции (индикаторы): ОПК-3, ПК-3

2. Установите соответствие названия сетей в зависимости от территориального расположения абонентских систем:

- | | |
|-----------------|------------------------------------|
| 1) Локальные | А) WAN — Wide Area Network |
| 2) Региональные | Б) LAN — Local Area Network |
| 3) Глобальные | В) MAN — Metropolitan Area Network |

Правильный ответ: 1-Б, 2-В, 3-А.

Компетенции (индикаторы): ОПК-3, ПК-3

Задания закрытого типа на установление правильной последовательности

Установите правильную последовательность.

Запишите правильную последовательность букв слева направо.

1. Для построения модели исходных данных надо выполнить действия в такой последовательности:

- А) разработка алгоритма,
- Б) анализ и интерпретация результатов,
- В) отладка и исполнение программы,
- Г) написание кода программы,
- Д) построение модели результата.

Правильный ответ: Д, А, Г, В, Б

Компетенции (индикаторы): ОПК-3, ПК-3

Задания открытого типа

Задания открытого типа на дополнение

Напишите пропущенное слово (словосочетание).

1. Существует _____ основных этапов разработки и исследование моделей на компьютере.

Правильный ответ: пять.

Компетенции (индикаторы): ОПК-3, ПК-3

2. Моделирование – это _____, состоящий в создании и исследовании моделей.

Правильный ответ: метод познания

Компетенции (индикаторы): ОПК-3, ПК-3

3. Описание глобальной компьютерной сети Интернет в виде системы взаимосвязанных объектов следует рассматривать как _____.

Правильный ответ: сетевую модель

Компетенции (индикаторы): ОПК-3, ПК-3

Задания открытого типа с кратким свободным ответом

Дайте ответ на вопрос.

1. Укажите один из методов реализации математической модели:

Правильный ответ: Алгебраические/ Приближенные/ Численные/ Имитационные

Компетенции (индикаторы): ОПК-3, ПК-3

2. Какой тип данных будет у итогового результата, если математическое выражение прибавляет вещественное число к целочисленному?

Правильный ответ: Вещественный/ вещественные данные

Компетенции (индикаторы): ОПК-3, ПК-3

Задания открытого типа с развёрнутым ответом

1. Найти с точностью 0.001 корни уравнения $x^{**2} \cdot \cos(2 \cdot x) + 1 = 0$ на отрезке $[0, \pi/2]$ методом половинного деления. Напишите код программы на Питоне.

Приведите полное решение.

Время выполнения – 60 мин.

Критерии оценивания: полное содержательное соответствие приведённому ниже решению.

Ожидаемый результат:

Решение:

```
import numpy as np
```

```
# Определяем функцию
```

```
def f(x):
    return x**2 * np.cos(2*x) + 1
```

```
# Метод половинного деления
```

```
def bisection_method(func, a, b, tol):
    if func(a) * func(b) >= 0:
        raise ValueError("Функция должна иметь разные знаки на концах отрезка [a, b].")
```

```

while (b - a) / 2.0 > tol:
    midpoint = (a + b) / 2.0
    if func(midpoint) == 0:
        return midpoint # Мы нашли корень
    elif func(a) * func(midpoint) < 0:
        b = midpoint # Корень в левой половине
    else:
        a = midpoint # Корень в правой половине

return (a + b) / 2.0 # Возвращаем среднюю точку как корень

# Задаем параметры
a = 0
b = np.pi / 2
tolerance = 0.001

# Находим корень
try:
    root = bisection_method(f, a, b, tolerance)
    print(f"Найденный корень: {root:.5f}")
except ValueError as e:
    print(e)

```

Компетенции (индикаторы): ОПК-3, ПК-3

2. Дано натуральное число n . Написать код программы на Питоне для вычисления:

$$1/1 + 1/2 + 1/3 + \dots + 1/n$$

Время выполнения – 40 мин.

Критерии оценивания: полное содержательное соответствие приведённому ниже решению.

Ожидаемый результат:

Решение:

```

def harmonic_sum(n):
    if n <= 0:
        raise ValueError("n должно быть натуральным числом.")

    total = 0.0
    for i in range(1, n + 1):
        total += 1 / i
    return total

```

Пример использования:

```
n = int(input("Введите натуральное число n: "))
result = harmonic_sum(n)
print(f"Сумма 1/1 + 1/2 + ... + 1/{n} = {result}")
```

Компетенции (индикаторы): ОПК-3, ПК-3

3. Используя приближенное интегрирование, найти приближенное значение π . Для этого вычислить площадь единичного круга, а затем увеличить ее в четыре раза. В алгоритме прямоугольников считать $n = 50$.

Время выполнения – 40 мин.

Критерии оценивания: полное содержательное соответствие приведённому ниже решению.

Ожидаемый результат:

Решение:

```
def approximate_pi(n):
    # Шаг интегрирования
    dx = 1.0 / n
    area = 0.0

    for i in range(n):
        # Высота прямоугольника
        x = (i + 0.5) * dx
        height = (1 - x**2) ** 0.5 # y = sqrt(1 - x^2)
        area += height * dx

    # Площадь единичного круга
    circle_area = area

    # Увеличиваем площадь в 4 раза для получения приближенного значения pi
    pi_approx = circle_area * 4
    return pi_approx

n = 50
pi_value = approximate_pi(n)
print(f"Приближенное значение Pi: {pi_value}")
```

Компетенции (индикаторы): ОПК-3, ПК-3

4. Задана функция $y(x)$ в виде

$X = 0, 1, 2, 3, 4, 5$

$Y = 3.1, 2.6, 4.7, 5.8, 10.1, 7.3$

С помощью алгоритма приближенного интегрирования (алгоритма трапеций)

вычислить площадь криволинейной трапеции.

Время выполнения – 35 мин.

Критерии оценивания: полное содержательное соответствие приведенному ниже решению.

Ожидаемый результат:

Решение:

```
# Заданные значения
X = [0, 1, 2, 3, 4, 5]
Y = [3.1, 2.6, 4.7, 5.8, 10.1, 7.3]

def trapezoidal_rule(x, y):
    n = len(x) - 1 # Количество трапеций
    area = 0.0
    for i in range(n):
        # Площадь трапеции: (b-a) * (f(a) + f(b)) / 2
        area += (x[i+1] - x[i]) * (y[i] + y[i+1]) / 2
    return area

# Вычисление площади
area = trapezoidal_rule(X, Y)
print(f'Приближенная площадь криволинейной трапеции: {area:.4f}')
```

Компетенции (индикаторы): ОПК-3, ПК-3

5. Напишите код программы на Python для вычисления дискретного преобразования Фурье (ДПФ) с использованием алгоритма БПФ (быстрого преобразования Фурье).

Время выполнения – 60 мин.

Критерии оценивания: полное содержательное соответствие приведённому ниже решению.

Ожидаемый результат:

Решение:

```
import numpy as np
import matplotlib.pyplot as plt

# Функция для вычисления ДПФ
def discrete_fourier_transform(signal):
    N = len(signal)
    dft = np.zeros(N, dtype=complex)
    for k in range(N):
        for n in range(N):
            dft[k] += signal[n] * np.exp(-2j * np.pi * k * n / N)
```

```

return dft

# Пример сигнала
t = np.linspace(0, 1, 100, endpoint=False) # Временная ось
signal = np.sin(2 * np.pi * 10 * t) + 0.5 * np.sin(2 * np.pi * 20 * t) # Сигнал

# Вычисление ДПФ
dft_result = discrete_fourier_transform(signal)

# Вывод результатов
frequencies = np.fft.fftfreq(len(signal), d=t[1]-t[0])
plt.subplot(2, 1, 1)
plt.plot(t, signal)
plt.title('Исходный сигнал')
plt.xlabel('Время')
plt.ylabel('Амплитуда')

plt.subplot(2, 1, 2)
plt.plot(frequencies, np.abs(dft_result))
plt.title('Дискретное преобразование Фурье')
plt.xlabel('Частота')
plt.ylabel('Амплитуда')

plt.tight_layout()
plt.show()

```

Компетенции (индикаторы): ОПК-3, ПК-3

6. Напишите код программы на Python для вейвлет-преобразования функции.
Время выполнения – 40 мин.

Критерии оценивания: полное содержательное соответствие приведённому ниже решению.

Ожидаемый результат:

Решение:

```

import numpy as np
import pywt
import matplotlib.pyplot as plt

# Создаем тестовую функцию (синусоида)
x = np.linspace(0, 1, 100)
y = np.sin(2 * np.pi * 5 * x) # Синусоида с частотой 5 Гц

# Выполняем вейвлет-преобразование

```

```

wavelet = 'haar' # Используем вейвлет Хаара
coeffs = pywt.wavedec(y, wavelet)

# Восстанавливаем сигнал из коэффициентов
reconstructed_y = pywt.waverec(coeffs, wavelet)

# Визуализируем оригинальный и восстановленный сигнал
plt.figure(figsize=(12, 6))
plt.subplot(2, 1, 1)
plt.plot(x, y, label='Оригинальный сигнал')
plt.title('Оригинальный сигнал')
plt.legend()

plt.subplot(2, 1, 2)
plt.plot(x, reconstructed_y, label='Восстановленный сигнал', color='orange')
plt.title('Восстановленный сигнал после вейвлет-преобразования')
plt.legend()

plt.tight_layout()
plt.show()

```

Компетенции (индикаторы): ОПК-3, ПК-3

7. Напишите код программы на Python для решения волнового уравнения для $u(x, t)$ методом сеток, $u(x, t)$ – функция, представляющая волну, а c – скорость распространения волны.

Время выполнения – 90 мин.

Критерии оценивания: полное содержательное соответствие приведённому ниже решению.

Ожидаемый результат:

Решение:

```

import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation

# Параметры
L = 10.0      # Длина области
T = 5.0       # Время симуляции
Nx = 100      # Количество пространственных узлов
Nt = 500      # Количество временных шагов
c = 1.0       # Скорость волны

# Дискретизация

```

```

dx = L / (Nx - 1) # Шаг по пространству
dt = T / Nt      # Шаг по времени

# Проверка условия стабильности
if dt > dx / c:
    raise ValueError("Слишком большой шаг времени, не удовлетворяет условию
стабильности.")

# Инициализация сетки
x = np.linspace(0, L, Nx)
u = np.zeros(Nx) # Текущее состояние
u_new = np.zeros(Nx) # Следующее состояние
u_old = np.zeros(Nx) # Предыдущее состояние

# Начальные условия (например, гауссовская волна)
u = np.exp(-((x - L/2)**2) / 0.1)
u_old = u.copy()

# Основной цикл по времени
frames = []
for n in range(1, Nt):
    for i in range(1, Nx - 1):
        u_new[i] = (2 * u[i] - u_old[i] +
                    (c * dt / dx)**2 * (u[i + 1] - 2 * u[i] + u[i - 1])))

    # Обновление старых значений
    u_old = u.copy()
    u = u_new.copy()

    # Сохраняем кадр для анимации
    frames.append(u.copy())

# Визуализация
fig, ax = plt.subplots()
line, = ax.plot(x, frames[0])
ax.set_ylim(-1, 1)
ax.set_title("Волновое уравнение")
ax.set_xlabel("x")
ax.set_ylabel("u")

def update(frame):
    line.set_ydata(frame)
    return line,

ani = FuncAnimation(fig, update, frames=frames, blit=True, interval=50)

```

plt.show()

Компетенции (индикаторы): ОПК-3, ПК-3

8. Напишите код решения волнового уравнения функции $u(x,y,t)$ на питоне. Решение волнового уравнения в двумерном пространстве можно реализовать с использованием численных методов (метод конечных разностей).

Время выполнения – 90 мин.

Критерии оценивания: полное содержательное соответствие приведённому ниже решению.

Ожидаемый результат:

Решение:

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation

# Параметры
Lx, Ly = 10.0, 10.0 # Размеры области
Nx, Ny = 100, 100 # Количество точек по x и y
dx = Lx / (Nx - 1) # Шаг по x
dy = Ly / (Ny - 1) # Шаг по y
c = 1.0 # Скорость волны
dt = 0.01 # Шаг по времени
T = 2.0 # Общее время симуляции

# Сеточная координатная сетка
x = np.linspace(0, Lx, Nx)
y = np.linspace(0, Ly, Ny)
X, Y = np.meshgrid(x, y)

# Начальные условия
u = np.sin(np.pi * X / Lx) * np.sin(np.pi * Y / Ly) # Начальное смещение
u_prev = u.copy() # Предыдущее состояние
u_next = np.zeros_like(u) # Следующее состояние

# Основной цикл
steps = int(T / dt)
for n in range(steps):
    # Обновление следующего состояния
    u_next[1:-1, 1:-1] = (2 * u[1:-1, 1:-1]
                          - u_prev[1:-1, 1:-1]
                          + (c * dt / dx)**2 * (u[1:-1, :-2] - 2 * u[1:-1, 1:-1] + u[1:-1, 2:])
                          + (c * dt / dy)**2 * (u[:-2, 1:-1] - 2 * u[1:-1, 1:-1] + u[2:, 1:-1]))
```

```

# Обновление значений для следующего шага
u_prev = u.copy()
u = u_next.copy()

# Визуализация результата
fig, ax = plt.subplots()
im = ax.imshow(u, extent=[0, Lx, 0, Ly], origin='lower', vmin=-1, vmax=1,
               cmap='viridis')

def update(frame):
    global u, u_prev, u_next
    u_next[1:-1, 1:-1] = (2 * u[1:-1, 1:-1]
                        - u_prev[1:-1, 1:-1]
                        + (c * dt / dx)**2 * (u[1:-1, :-2] - 2 * u[1:-1, 1:-1] + u[1:-1, 2:])
                        + (c * dt / dy)**2 * (u[:-2, 1:-1] - 2 * u[1:-1, 1:-1] + u[2:, 1:-1]))

    u_prev = u.copy()
    u = u_next.copy()
    im.set_array(u)
    return [im]

ani = FuncAnimation(fig, update, frames=steps, interval=50)
plt.colorbar(im)
plt.title('Решение волнового уравнения')
plt.xlabel('x')
plt.ylabel('y')
plt.show()

```

Компетенции (индикаторы): ОПК-3, ПК-3

Экспертное заключение

Представленный фонд оценочных средств (далее - ФОС) по дисциплине «Математическое моделирование сигналов и систем» соответствует требованиям ФГОС ВО.

Предлагаемые формы и средства текущего и промежуточного контроля адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 12.04.01 Приборостроение.

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины представлены в полном объеме.

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

Разработанный и представленный для экспертизы фонд оценочных средств рекомендуется к использованию в процессе подготовки обучающихся по указанному направлению 12.04.01 Приборостроение.

Председатель
учебно-методической комиссии
института



Яременко С.П.

Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)