Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Теория систем и системный анализ

ОТЧЕТ

Работа 2. Измерение и оценка свойств системы. Классификация

Проверил
Преподаватель
_____ Кочергин М.И.

Выполнил
Студент гр. 571-2
_____Вьюгин К.В.

Томск
2022

Вариант 5 (Пианино)

1)

1.1) Таблица 1.1 - Измерение свойств пианино с помощью номинальной шкалы

Объект	Производитель (модель)	Габариты	Цвет	Вес
x1	Pearl River (EU118)	118 * 150 * 59,8 см	Красное дерево	248 кг
x2	Kawai (K200 NKL M/ PEP)	114 * 149 * 57 см	Черный	208 кг
x3	Ritmuller (RS125)	125 * 152 * 63 см	Черный	260 кг
x4	Yamaha (YUS1 PM)	128 * 160 * 69 см	Красное дерево	259 кг
x5	Petrof P (135K1)	155 * 145 * 70 см	Черный	282 кг

Таблица 1.2 – Сравнение совпадений свойств пианино

Свойство	Символ Кронекера									
	δ12	δ13	δ14	δ15	δ23	δ24	δ25	δ34	δ35	δ45
Производитель (модель)	0	0	0	0	0	0	0	0	0	0
Габариты	0	0	0	0	0	0	0	0	0	0
Цвет	0	0	1	0	1	0	1	0	1	0
Вес	0	0	0	0	0	0	0	0	0	0

Найдем частоту и моду для каждого свойства:

Производитель (модель): Моды **нет**

Pearl River (EU118): $n=1/5$ **Kawai (K200 NKL M/ PEP):** $n=1/5$ **Ritmuller (RS125):** $n=1/5$ **Yamaha (YUS1 PM):** $n=1/5$ **Petrof P (135K1):** $n=1/5$

Габариты: Моды **нет**

118 * 150 * 59,8 см : n=$^{1}/_{5}$ **114 * 149 * 57 см :** n=$^{1}/_{5}$ **125 * 152 * 63 см :** n=$^{1}/_{5}$ **128 * 160 * 69 см :** n=$^{1}/_{5}$

155 * 145 * 70 см : n=$^{1}/_{5}$

Цвет: Мода – класс **Черный**

Красное дерево: n=$^{2}/_{5}$ **Черный:** n=$^{3}/_{5}$

Вес: Моды **нет**

248 кг : n=$^{1}/_{5}$ **208 кг :** n=$^{1}/_{5}$ **260 кг :** n=$^{1}/_{5}$ **259 кг :** n=$^{1}/_{5}$ **282 кг :** n=$^{1}/_{5}$

1.2)Таблица 1.3 – Измерение свойств пианино с помощью ранговой шкалы

Свойства	Ранги					Медиана
	x1	x2	x3	x4	x5	
Средний отзыв	4	3	5	1	2	x2
Привлекательность	5	2	4	3	1	x4

1.3) Таблица 1.4 – Измерение свойств с помощью шкал интервалов и отношений

Свойство	x1	x2	x3	x4	x5
Цена	316 000,00 ₽	810 000,00 ₽	500 000,00 ₽	1 300 000,00 ₽	2 321 000,00 ₽
Гарантия	6 месяцев	12 месяцев	6 месяцев	12 месяцев	12 месяцев

2) Вариант-5

Имеется 8 стран, каждую из которых можно охарактеризовать рядом параметров: название, континент, суммарный ВВП, внешний долг, плотность населения.

Характеристики стран можно представить в виде таблицы:

Номер	Тип	ВВП, млрд $	Долг, трл. $	Плотность насел-я, чел./кв. км.
1	Развитая	0,305	0,627	128
2	Развивающаяся	0,217	0,028	80
3	Развивающаяся	0,285	0,043	29,8
4	Развитая	3,306	5,624	230
5	Развитая	0,729	0,31	100
6	Развивающаяся	2,024	4,07	23,6
7	Развитая	1,477	0,396	8,4
8	Развивающаяся	0,354	0,074	40
9	?	14,624	16,014	32
10	?	1,22	1,376	2,8

Определить, в каких шкалах заданы характеристики перечисленных стран.

Составить программу, определяющую тип стран 9 и 10.

Использовать метод k-ближайших соседей при

А) k = 1, Б) k = 3

Результат: При k = 1:

9 - развивающаяся

10 - развитая

При k = 3:

9 - развивающаяся

10 - развивающаяся

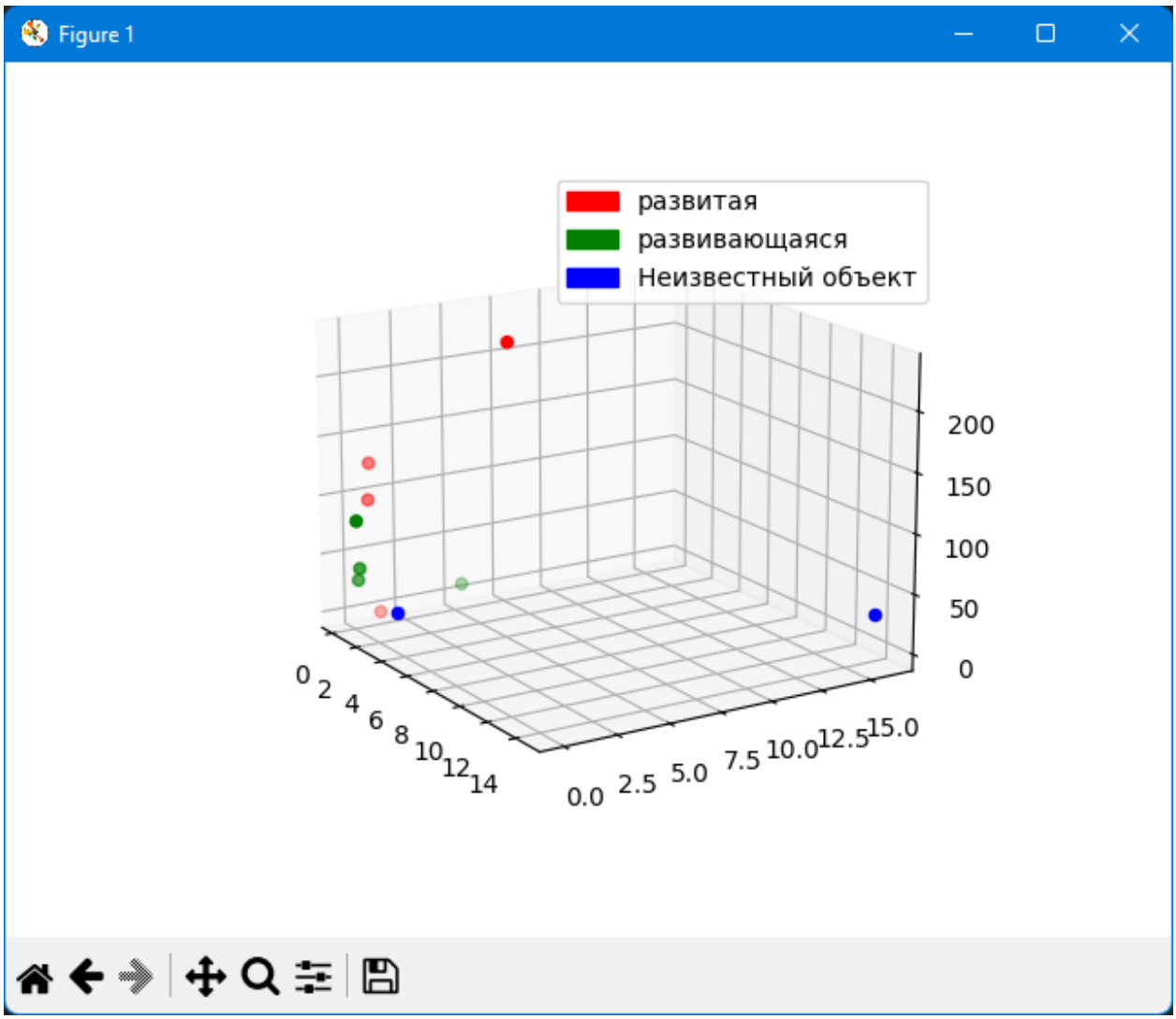
Код:

```
"""Импортируем нужные библиотеки"""
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches

"""Для отображения графика"""
matplotlib.use('TkAgg')

"""Функция поиска минимального расстояния между точками (1 сосед)"""
def minDistance1(GPD, debt, populationDensity, X, Y, Z):
    minimum = np.sqrt(((X[0] - GPD) * (X[0] - GPD)) +
                      ((Y[0] - debt) * (Y[0] - debt)) +
                      ((Z[0] - populationDensity) * (Z[0] -
populationDensity)))

    for i in range(X.size):
        value = np.sqrt(((X[i] - GPD) * (X[i] - GPD)) +
                        ((Y[i] - debt) * (Y[i] - debt)) +
                        ((Z[i] - populationDensity) * (Z[i] -
populationDensity)))

        minimum = value if value < minimum else minimum
    return minimum

"""Функция поиска минимального расстояния между точками (3 соседа)"""
def minDistance3(GPD, debt, populationDensity, X1, Y1, Z1, X2, Y2, Z2):
    minimum = np.array([])
    valueOfDeveloped = np.array([])

    for i in range(X1.size):
        minimum = np.append(minimum, np.sqrt(((X1[i] - GPD) * (X1[i] - GPD))
+
                                             ((Y1[i] - debt) * (Y1[i] -
debt)) +
                                             ((Z1[i] - populationDensity) *
(Z1[i] - populationDensity))))
        valueOfDeveloped = np.append(valueOfDeveloped, "Развитая")

    for i in range(X2.size):
        minimum = np.append(minimum, np.sqrt(((X2[i] - GPD) * (X2[i] - GPD))
+
                                             ((Y2[i] - debt) * (Y2[i] -
debt)) +
                                             ((Z2[i] - populationDensity) *
(Z2[i] - populationDensity))))
        valueOfDeveloped = np.append(valueOfDeveloped, "Развивающаяся")

    indices = minimum.argsort()
    valueOfDeveloped = valueOfDeveloped[indices][0:3]

    if (np.count_nonzero(valueOfDeveloped == "Развитая") >
```

```
np.count_nonzero(valueOfDeveloped == "Развивающаяся")):
        return "Развитая"
    else:
        return "Развивающаяся"

"""Данные на вход"""
countryTypes = np.array(["Развитая", "Развивающаяся", "Развивающаяся",
"Развитая", "Развитая", "Развивающаяся", "Развитая", "Развивающаяся"])
ountrysGDP = np.array([0.305, 0.217, 0.285, 3.306, 0.729, 2.024, 1.477,
0.354])
CountrysDebt = np.array([0.627, 0.028, 0.043, 5.624, 0.310, 4.07, 0.396,
0.074])
populationDensityOfCountry = np.array([128, 80, 29.8, 230, 100, 23.6, 8.4,
40])

"""Развитые"""
X1 = np.array([])
Y1 = np.array([])
Z1 = np.array([])

"""Развивающиеся"""
X2 = np.array([])
Y2 = np.array([])
Z2 = np.array([])

"""Отбираем страны на развитые и на развивающиеся"""
for i in range(len(countryTypes)):
    if countryTypes[i] == "Развитая":
        X1 = np.append(X1, ountrysGDP[i])
        Y1 = np.append(Y1, CountrysDebt[i])
        Z1 = np.append(Z1, populationDensityOfCountry[i])
    else:
        X2 = np.append(X2, ountrysGDP[i])
        Y2 = np.append(Y2, CountrysDebt[i])
        Z2 = np.append(Z2, populationDensityOfCountry[i])

fig = plt.figure()
ax = fig.add_subplot(projection='3d')

"""Данные стран с неизвестными типами"""
countryTypes_unknown = np.array([9, 10])
GPD_unknown = np.array([14.624, 1.220])
debt_unknown = np.array([16.014, 1.376])
populationDensity_unknown = np.array([32, 2.8])

print("При k = 1:")

"""Проверяем каждую страну на принадлежность к типам ("Развитая",
"Развивающаяся") при (k = 1)"""
for i in range(2):
    developed = minDistance1(GPD_unknown[i], debt_unknown[i],
populationDensity_unknown[i], X1, Y1, Z1)
    developing = minDistance1(GPD_unknown[i], debt_unknown[i],
populationDensity_unknown[i], X2, Y2, Z2)

ax.scatter(GPD_unknown[i],debt_unknown[i],populationDensity_unknown[i],color=
'blue')

    if developed < developing:
        print(str(countryTypes_unknown[i]) + " - развитая")
    else:
        print(str(countryTypes_unknown[i]) + " - развивающаяся")

print("При k = 3:")

"""Проверяем каждую страну на принадлежность к типам ("Развитая",
```

```
"Развивающаяся") при (k = 3)"""
for i in range(2):
    answer = minDistance3(GPD_unknown[i], debt_unknown[i],
populationDensity_unknown[i], X1, Y1, Z1, X2, Y2, Z2)

    if (answer == "Развитая"):
        print(str(countryTypes_unknown[i]) + " - развитая")
    else:
        print(str(countryTypes_unknown[i]) + " - развивающаяся")

"""Выводим график"""
ax.scatter(X1,Y1,Z1,color='red')
ax.scatter(X2,Y2,Z2,color='green')
plt.legend(handles=[mpatches.Patch(color='red', label='развитая'),
                    mpatches.Patch(color='green', label='развивающаяся'),
                    mpatches.Patch(color='blue', label='Неизвестный
объект')])
plt.show()
```

График:



Контрольные вопросы:

1)Теория измерений изучает закономерности хранения, воспроизведения, передачи, получения, обработки, использования, а также оценки качества (точности и достоверности) измерительной информации.

2)Шкала – это знаковая система. (система с надлежащими отношениями меж символами.)

3) Номинальная, интервальная, порядковая, абсолютная и отношений.

4) Шкала интервалов, отношений, абсолютная.

5) Формируется матрица предоставленной номинальной шкалы, где и в строчках, и в столбцах заданы объекты наблюдаемого качества. Вслед за тем ориентируется аксессуар всякой ячейки матрицы. На базе предоставленной матрицы основывается свежая, по формуле

$$d_{ij}^{H} = \frac{1}{N}\sum_{k=1}^{N} d_k(i,j)$$

, где dij – вещество ячейки, N – численность объектов, i – номер *dk(i,j)* строчки, j – номер столбца, - мерило различия объектов в строчках i и j сравнительно объекта k

6) В начале возводятся матрицы расстояний меж объектами по всем требуемым свойствам, а вслед за тем сплошное расстояние рассчитывается как корень суммы квадратов этих свойств.