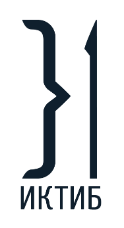
МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждения высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Кафедра математического обеспечения и применения ЭВМ



**ЛАБОРАТОРНАЯ РАБОТА № 2**

**ОТЧЁТ**

по дисциплине

**«Разработка прикладных решений на языке Python»**

Выполнил:

Студент группы

КТмо2-16

|  |  |  |
| --- | --- | --- |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Руденко К. Д. |
|  | *подпись* |  |

Проверил:

Доцент кафедры САиТ, к.т.н.

|  |  |  |
| --- | --- | --- |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Лапшин В. С. |
|  | *подпись* |  |

Оценка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.

**2 лабораторная работа.**

**Руденко Константин Дмитриевич – 5 вариант**

**Часть 1  
задания 3, 1, 6, 8**

**Задача № 1**

Изображение выглядит как текст, снимок экрана, Шрифт, алгебра

Содержимое, созданное искусственным интеллектом, может быть неверным.

Реализация:

print("5 вариант")  
print("1 задача")  
  
x = int(input("Введите 1-ое целое число x = ")) # 1-ое число целое число  
y = int(input("Введите 2-ое целое число y = ")) # 2-ое число целое число  
  
# лямбда выражение которое проверяет вещественое ли число ? в случае True округляет до 2 знаков после запятой  
round\_if\_float = lambda now: round(now, 2) if isinstance(now, float) else now  
  
# Арифметические операции  
# +, -, \*, /, //, %, \*\*  
print("Арифметические операции")  
print("+, -, \*, /, //, %, \*\*")  
# +  
print(f"{x} + {y} = {x + y}")  
# -  
print(f"{x} - {y} = {x - y}")  
print(f"{y} - {x} = {y - x}")  
# \*  
print(f"{x} \* {y} = {x \* y}")  
# /  
print(f"{x} / {y} = {round\_if\_float(x / y)}")  
print(f"{y} / {x} = {round\_if\_float(y / x)}")  
# //  
print(f"{x} // {y} = {x // y}")  
print(f"{y} // {x} = {y // x}")  
# %  
print(f"{x} % {y} = {x % y}")  
print(f"{y} % {x} = {y % x}")  
# \*\*  
print(f"{x}\*\*{y} = {x\*\*y}")  
print(f"{y}\*\*{x} = {y\*\*x}")  
  
# Операции сравнения  
# <, <=, >, >=, !=, ==  
print("Операции сравнения")  
print("<, <=, >, >=, !=, ==")  
# <  
print(f"{x} < {y} = {x < y}")  
print(f"{y} < {x} = {y < x}")  
# <=  
print(f"{x} <= {y} = {x <= y}")  
print(f"{y} <= {x} = {y <= x}")  
# >  
print(f"{x} > {y} = {x > y}")  
print(f"{y} > {x} = {y > x}")  
# >=  
print(f"{x} >= {y} = {x >= y}")  
print(f"{y} >= {x} = {y >= x}")  
# !=  
print(f"{x} != {y} = {x != y}")  
# ==  
print(f"{x} == {y} = {x == y}")

C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\.venv\Scripts\python.exe C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\laba2\Task1.py

5 вариант

1 задача

Введите 1-ое целое число x = 2

Введите 2-ое целое число y = 3

Арифметические операции

+, -, \*, /, //, %, \*\*

2 + 3 = 5

2 - 3 = -1

3 - 2 = 1

2 \* 3 = 6

2 / 3 = 0.67

3 / 2 = 1.5

2 // 3 = 0

3 // 2 = 1

2 % 3 = 2

3 % 2 = 1

2\*\*3 = 8

3\*\*2 = 9

Операции сравнения

<, <=, >, >=, !=, ==

2 < 3 = True

3 < 2 = False

2 <= 3 = True

3 <= 2 = False

2 > 3 = False

3 > 2 = True

2 >= 3 = False

3 >= 2 = True

2 != 3 = True

2 == 3 = False

Process finished with exit code 0

**Задача № 3**

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Реализация:

print("5 вариант")  
print("3 задача")  
  
r1 = float(input("Введите 1-ое сопротивление R1 = ")) # 1-ое число целое число  
r2 = float(input("Введите 2-ое сопротивление R2 = ")) # 2-ое число целое число  
  
print("Цепь подключена последовательно")  
  
round\_if\_float = lambda sum\_r: round(sum\_r, 1) if isinstance(sum\_r, float) else sum\_r  
  
sum\_r = round\_if\_float(r1 + r2)  
  
print(f"Общее сопротивление цепи: {r1} + {r2} = {sum\_r} Ом")

# Результат работы программы:

C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\.venv\Scripts\python.exe C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\laba2\Task3.py

5 вариант

3 задача

Введите 1-ое сопротивление R1 = 2.4212

Введите 2-ое сопротивление R2 = 6.21325

Цепь подключена последовательно

Общее сопротивление цепи: 2.4212 + 6.21325 = 8.6 Ом

Process finished with exit code 0

**Задача № 6**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

print("5 вариант")  
print("6 задача")  
  
a = int(input("Введите коэффициент a: "))  
b = int(input("Введите коэффициент b: "))  
  
# Ввод границ отрезка  
m = int(input("Введите левую границу отрезка m: "))  
n = int(input("Введите правую границу отрезка n: "))  
  
round\_if\_float = lambda x: round(x, 1) if isinstance(x, float) else x  
print("Нахождение корня уравнения")  
if m <= n:  
 if a == 0:  
 if b == 0:  
 print("Уравнение имеет бесконечно много решений")  
 else:  
 print("Уравнение не имеет решений")  
 else:  
 x = round\_if\_float(-b / a)  
 in\_interval = m <= x <= n  
  
 print(f"Решение уравнения {a} \* x + {b} = 0 \n"  
 f"Ответ x = {x}")  
 print(f"Попадает ли {x} в отрезок: [{m}; {n}]? {'✓ Да' if in\_interval else '✗ Нет'}!")  
else:  
 print("Левая граница отрезка не может быть больше правой")

# Результат работы программы:

C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\.venv\Scripts\python.exe C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\laba2\Task6.py

5 вариант

6 задача

Введите коэффициент a: -1

Введите коэффициент b: 1

Введите левую границу отрезка m: -5

Введите правую границу отрезка n: 6

Нахождение корня уравнения

Решение уравнения -1 \* x + 1 = 0

Ответ x = 1.0

Попадает ли 1.0 в отрезок: [-5; 6]? ✓ Да!

Process finished with exit code 0

**Задача № 8**

**Изображение выглядит как текст, чек, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.**

print("5 вариант")  
print("8 задача")  
  
# Ввод государства и его столицы  
country = input("Введите название государства: ")  
capital = input("Введите название столицы: ")  
  
# Вывод государства и его столицы  
print(f"Государство — {country}, столица — {capital}")

# Результат работы программы:

C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\.venv\Scripts\python.exe C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\laba2\Task8.py

5 вариант

8 задача

Введите название государства: Россия

Введите название столицы: Москва

Государство — Россия, столица — Москва

Process finished with exit code 0

Ссылка на гитхаб: [all/Py3semestr2025RudenkoKTmo2-16/laba2 at master · Kosten-73/all](https://github.com/Kosten-73/all/tree/master/Py3semestr2025RudenkoKTmo2-16/laba2)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 1 – 1 задание

Изображение выглядит как текст, снимок экрана, программное обеспечение, компьютер

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 2 – 3 задание

Изображение выглядит как текст, снимок экрана, программное обеспечение, компьютер

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 3 – 6 задание

Изображение выглядит как снимок экрана, текст, программное обеспечение, компьютер

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 4 – 8 задание

Изображение выглядит как текст, снимок экрана, программное обеспечение, компьютер

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 5 – Варианты

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 6 – Гитхаб

Часть 2

1. Скалярные типы данных: числа, логический тип, NoneType. Определение, основные операции.

**Ответ**: Скалярные типы данных – это типы данных, которые содержат одно значение.

**Числа** могут быть целые (int), например 1, -4 или в других СС 0b1111 (15), -0b1111 (-15), вещественные (float) например (2.31, -431.31, 3.14 – операции +, -, \*, /, //, %, \*\*

**Логические** это булева алгебра. То есть переменная может быть: истина ли ложь (1 – истина - True, 0 – ложь - False). С помощью этих операций можно делать различные ветвление, операции - <, <=, >, >=, !=, ==, or, and, not, not in

**NoneType** – переменная или объект у которого отсутствует значение. То есть никак его не определили. Частая ошибка NPE

a = None

a is None

1. Отображения (словарь): определение, основные операции. Где целесообразно применять словари?

**Ответ:** Словарь (dict) – это коллекция, которую можно изменять и в ней могут различные типы данных и хранятся они в виде пар **ключ**: **значение.** Ключи – уникальны и не изменяются (одного типа), а значение любого типа.

d1 = {'a': 1, 'b': 2}

d2 = {'b': 3, 'c': 4}

эти словари можно объединить d = d1 | d2

{'a': 1, 'b': 3, 'c': 4}

Сравнить ==, !=

d1 != d2 - True

d1.k – d2.k; d2.k – d1.k

Словари применяются в json, подсчете элементов – допустим часто встречания букв в текст hello {h: 1, e: 1, l: 2, o: 1}, кеширование, в прикладных задачах это мапинг данных. Преимущество в быстром доступе чтобы получить значение по ключу асимптотическая сложность = 1 (O(1))

**Задание**

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

*Изображение выглядит как текст, меню, документ, снимок экрана

Автоматически созданное описание*

Решение:

# База данных книжного магазина  
db = [  
 {  
 "title": "Мастер и Маргарита",  
 "price": 850.21,  
 "quantity": 15,  
 "authors": ["Михаил Булгаков"],  
 "isbn": "3213-213-1-4431-341",  
 "rating": 4.8,  
 "hardcover": True,  
 "genres": ["роман", "фантастика", "сатира"]  
 },  
 {  
 "title": "Библия",  
 "price": 9999.99,  
 "quantity": 73,  
 "authors": ["Апостол1", "Апостол2"],  
 "isbn": "111-1-1-1-1-111",  
 "rating": 5.0,  
 "hardcover": True,  
 "genres": ["основано на реальных событиях", "теология", "вера"]  
 },  
 {  
 "title": "Преступление и наказание",  
 "price": 380.50,  
 "quantity": 8,  
 "authors": ["Федор Достоевский"],  
 "isbn": "111-1-1-1-1-111",  
 "rating": 4.9,  
 "hardcover": False,  
 "genres": ["роман", "психология", "классика"]  
 },  
 {  
 "title": "Питон для начинающих",  
 "price": 890.75,  
 "quantity": 25,  
 "authors": ["Иван Петров", "Мария Сидорова", "Алексей Кузнецов"],  
 "isbn": "412-214-21-4-21-4",  
 "rating": 4.5,  
 "hardcover": True,  
 "genres": ["программирование", "учебник", "компьютеры"]  
 },  
 {  
 "title": "1984",  
 "price": 520.25,  
 "quantity": 12,  
 "authors": ["Джордж Оруэлл"],  
 "isbn": "532-523-5-235-235",  
 "rating": 4.7,  
 "hardcover": True,  
 "genres": ["антиутопия", "политика", "фантастика"]  
 }  
]  
  
print("Содержимое базы данных книжного магазина ({} книги):".format(len(db)))  
  
for book\_i in range(len(db)):  
 print(f"{book\_i + 1}.")  
 print("Название: {}".format(db[book\_i]["title"]))  
 print("Цена: {} руб.".format(db[book\_i]["price"]))  
 print("Количество в наличии: {}".format(db[book\_i]["quantity"]))  
 print("Авторы: {}".format(", ".join(db[book\_i]["authors"])))  
 print("ISBN: {}".format(db[book\_i]["isbn"]))  
 print("Рейтинг: {}/5".format(db[book\_i]["rating"]))  
 print("Твердый переплет: {}".format(db[book\_i]["hardcover"]))  
 print("Жанры: {} \n".format(", ".join(db[book\_i]["genres"])))  
  
  
# --------------  
# Пример вывода:  
# --------------  
# C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\.venv\Scripts\python.exe C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\laba2\Part2.py  
# Содержимое базы данных книжного магазина (5 книги):  
# 1.  
# Название: Мастер и Маргарита  
# Цена: 850.21 руб.  
# Количество в наличии: 15  
# Авторы: Михаил Булгаков  
# ISBN: 3213-213-1-4431-341  
# Рейтинг: 4.8/5  
# Твердый переплет: True  
# Жанры: роман, фантастика, сатира  
#  
# 2.  
# Название: Библия  
# Цена: 9999.99 руб.  
# Количество в наличии: 73  
# Авторы: Апостол1, Апостол2  
# ISBN: 111-1-1-1-1-111  
# Рейтинг: 5.0/5  
# Твердый переплет: True  
# Жанры: основано на реальных событиях, теология, вера  
#  
# 3.  
# Название: Преступление и наказание  
# Цена: 380.5 руб.  
# Количество в наличии: 8  
# Авторы: Федор Достоевский  
# ISBN: 111-1-1-1-1-111  
# Рейтинг: 4.9/5  
# Твердый переплет: False  
# Жанры: роман, психология, классика  
#  
# 4.  
# Название: Питон для начинающих  
# Цена: 890.75 руб.  
# Количество в наличии: 25  
# Авторы: Иван Петров, Мария Сидорова, Алексей Кузнецов  
# ISBN: 412-214-21-4-21-4  
# Рейтинг: 4.5/5  
# Твердый переплет: True  
# Жанры: программирование, учебник, компьютеры  
#  
# 5.  
# Название: 1984  
# Цена: 520.25 руб.  
# Количество в наличии: 12  
# Авторы: Джордж Оруэлл  
# ISBN: 532-523-5-235-235  
# Рейтинг: 4.7/5  
# Твердый переплет: True  
# Жанры: антиутопия, политика, фантастика  
#  
#  
# Process finished with exit code 0

Часть 3

**Задача 1.**

**Изображение выглядит как текст, Шрифт, белый, чек

Содержимое, созданное искусственным интеллектом, может быть неверным.**

# Задача 1  
a = int(input("Введите высоту форточки (a, см): "))  
b = int(input("Введите ширину форточки (b, см): "))  
d = int(input("Введите диаметр головы Васи (d, см): "))  
if d <= a - 2 and d <= b - 2:  
 print("Вася прирожденный форточник!)")  
else:  
 print("Васе прийдется устроиться на работу - он в форточку не пролез")  
  
# Или  
if d <= min(a - 2, b - 2):  
 print("Вася прирожденный форточник!)")  
else:  
 print("Васе прийдется устроиться на работу - он в форточку не пролез")

C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\.venv\Scripts\python.exe C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\laba2\3\Task3-1.py

Введите высоту форточки (a, см): 20

Введите ширину форточки (b, см): 25

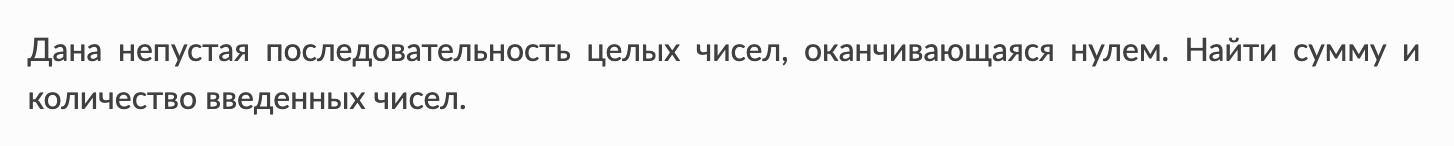
Введите диаметр головы Васи (d, см): 18

Вася прирожденный форточник!)

Вася прирожденный форточник!)

Process finished with exit code 0

**Задача 3.**

****

# Задача 1  
a = int(input("Введите целое число: "))  
s = 0  
k = 0  
while (a != 0):  
 s += a  
 k += 1  
 a = int(input("Введите целое число: "))  
print(f"Сумма = {s} и количество введенных чисел из последовательности = {k}")  
  
# C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\.venv\Scripts\python.exe C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\laba2\3\Task3-3.py  
# Введите целое число: 1  
# Введите целое число: 3  
# Введите целое число: 0  
# Сумма = 4 и количество введенных чисел из последовательности = 2  
#  
# Process finished with exit code 0

**Задача 6.**

**Изображение выглядит как текст, Шрифт, белый, чек

Содержимое, созданное искусственным интеллектом, может быть неверным.**

# Задача 1  
a = int(input("Введите целое число a: "))  
b = int(input("Введите целое число b: "))  
# Пункт 1  
print("Числа от минимального до максимального:")  
for i in range(min(a, b), max(a, b) + 1):  
 print(i, end=' ')  
# Пункт 2  
print("\nЧисла от максимального до минимального столбиком:")  
for i in range(max(a, b), min(a, b) - 1, - 1):  
 print(i)  
  
# C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\.venv\Scripts\python.exe C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\laba2\3\Task3-6.py   
# Введите целое число a: 3  
# Введите целое число b: 7  
# Числа от минимального до максимального:  
# 3 4 5 6 7   
# Числа от максимального до минимального столбиком:  
# 7  
# 6  
# 5  
# 4  
# 3  
#   
# Process finished with exit code 0

C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\.venv\Scripts\python.exe C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\laba2\3\Task3-6.py

Введите целое число a: 7

Введите целое число b: 3

Числа от минимального до максимального:

3 4 5 6 7

Числа от максимального до минимального столбиком:

7

6

5

4

3

Process finished with exit code 0

**Задача 8.**

**Изображение выглядит как текст, Шрифт, снимок экрана, алгебра

Содержимое, созданное искусственным интеллектом, может быть неверным.**

n = int(input("Введите колличество учеников: "))  
  
k\_man, h\_man = 0, 0  
k\_woman, h\_woman = 0, 0  
  
for i in range(n):  
 height = float(input("Введите рост ученика (см): "))  
 if height < 0:  
 k\_man += 1  
 h\_man += abs(height)  
 else:  
 k\_woman += 1  
 h\_woman += height  
h\_man /= k\_man  
h\_woman /= k\_woman  
all\_k = k\_man + k\_woman  
all\_h = (h\_man + h\_woman) / 2  
  
print(f"Количество мужчин: {k\_man}, Средний рост мужчин: {h\_man:.1f}, "  
 f"Количество женщин: {k\_woman}, Средний рост женщин: {h\_woman:.1f}, "  
 f"Средний рост всех учеников: {all\_h:.1f}, "  
 f"Количество учеников: {all\_k}")

n = int(input("Введите колличество учеников: "))  
  
k\_man, h\_man = 0, 0  
k\_woman, h\_woman = 0, 0  
  
for i in range(n):  
 height = float(input("Введите рост ученика (см): "))  
 if height < 0:  
 k\_man += 1  
 h\_man += abs(height)  
 else:  
 k\_woman += 1  
 h\_woman += height  
h\_man /= k\_man  
h\_woman /= k\_woman  
all\_k = k\_man + k\_woman  
all\_h = (h\_man + h\_woman) / 2  
  
print(f"Количество мужчин: {k\_man}, Средний рост мужчин: {h\_man:.1f}, "  
 f"Количество женщин: {k\_woman}, Средний рост женщин: {h\_woman:.1f}, "  
 f"Средний рост всех учеников: {all\_h:.1f}, "  
 f"Количество учеников: {all\_k}")  
  
# C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\.venv\Scripts\python.exe C:\Users\korudenko\PycharmProjects\Py3semestr2025RudenkoKTmo2-16\laba2\3\Task3-8.py   
# Введите колличество учеников: 5  
# Введите рост ученика (см): 1321.1  
# Введите рост ученика (см): -213.131232  
# Введите рост ученика (см): 321  
# Введите рост ученика (см): 41  
# Введите рост ученика (см): 213.1231  
# Количество мужчин: 1, Средний рост мужчин: 213.1, Количество женщин: 4, Средний рост женщин: 474.1, Средний рост всех учеников: 343.6, Количество учеников: 5  
#   
# Process finished with exit code 0