

1. ПРАКТИЧЕСКАЯ РАБОТА №1.

1.1 Задание 1

Задание 1 представлено на рисунке 1.1.

2. Написать программу, которая вычисляет площадь фигуры, параметры которой подаются на вход. Фигуры, которые подаются на вход: треугольник, прямоугольник, круг. Результатом работы является словарь, где ключ – это название фигуры, а значение – это площадь.

Рисунок 1.1 — Задание 1

Код задания представлен в листинге 1.1.

Листинг 1.1 — Код Задания 1

```
tri_a,h = input("Введите сторону и высоту треугольника: ").split()
sq_a = input("Введите сторону квадрата: ")
R = input("Введите радиус окружности: ")
squares = {"Треугольник": 1/2*int(tri_a)*int(h), "Квадрат": int(sq_a)**2, "Окружность": 3.14*int(R)**2}
print(squares)
```

Результат выполнения Задания 1 представлен на рисунке 1.2

```
● maga@Magomeds-MacBook-Pro ~ % cd "/Users/maga/" && python3 temp.py
Введите сторону и высоту треугольника: 4 6
Введите сторону квадрата: 10
Введите радиус окружности: 5
{'Треугольник': 12.0, 'Квадрат': 100, 'Окружность': 78.5}
○ maga@Magomeds-MacBook-Pro ~ %
```

Рисунок 1.2 — Результат выполнения Задания 1

1.2 Задание 2

Задание 2 представлено на рисунке 1.3.

3. Написать программу, которая на вход получает два числа и операцию, которую к ним нужно применить. Должны быть реализованы следующие операции: +, -, /, //, abs – модуль, pow или ** – возведение в степень.

Рисунок 1.3 — Задание 2

Код задания представлен в листинге 1.2.

Листинг 1.2 — Код Задания 2

```
a, b, operator = input("Введите a, b и оператор: ").split()
```

```

a = int(a)
b = int(b)

if operator == "+":
    print(a+b)
elif operator == "-":
    print(a-b)
elif operator == "*":
    print(a*b)
elif operator == "/":
    print(a/b)
elif operator == "//":
    print(a//b)
elif operator == "**" or operator == "pow":
    print(a**b)
elif operator == "abs":
    print(abs(a))

```

Результат выполнения Задания 2 представлен на рисунке 1.4

```

● maga@Magomeds-MacBook-Pro ~ % cd "/Users/maga/" && python3 temp.py
Введите a, b и оператор: 1 2 +
3
○ maga@Magomeds-MacBook-Pro ~ % 

```

Рисунок 1.4 — Результат выполнения Задания 2

1.3 Задание 3

Задание 3 представлено на рисунке 1.5.

Напишите программу, которая считывает с консоли числа (по одному в строке) до тех пор, пока сумма введённых чисел не будет равна 0 и после этого выводит сумму квадратов всех считанных чисел.

Рисунок 1.5 — Задание 3

Код задания представлен в листинге 1.3.

Листинг 1.3 — Код Задания 3

```

sum = int(input())
square_sum = sum**2
while sum != 0:
    sum += int(input())
    square_sum += sum**2
print(square_sum)

```

Результат выполнения Задания 3 представлен на рисунке 1.6

```

maga@Magomeds-MacBook-Pro tests % cd "/Us
7
8
10
-25
899
maga@Magomeds-MacBook-Pro tests % █

```

Рисунок 1.6 — Результат выполнения Задания 3

1.4 Задание 4

Задание 4 представлено на рисунке 1.7.

Напишите программу, которая выводит последовательность чисел, длиной N, где каждое число повторяется столько раз, чему оно равно. На вход программе передаётся неотрицательное целое число N. Например, если N = 7, то программа должна вывести 1 2 2 3 3 3 4. Вывод элементов списка через пробел – `print(*list)`.

Рисунок 1.7 — Задание 4

Код задания представлен в листинге 1.4.

Листинг 1.4 — Код Задания 4

```

N = int(input())
a=[]
for i in range(N):
    for k in range(i):
        a.append(i)
    if len(a) >= N:
        break
print(*a[:N])

```

Результат выполнения Задания 4 представлен на рисунке 1.8

```

maga@Magomeds-MacBook-Pro tests % cd "
9
1 2 2 3 3 3 4 4 4
maga@Magomeds-MacBook-Pro tests % █

```

Рисунок 1.8 — Результат выполнения Задания 4

1.5 Задание 5

Задание 5 представлено на рисунке 1.9.

Даны два списка:

```

A = [1, 2, 3, 4, 2, 1, 3, 4, 5, 6, 5, 4, 3, 2]
B = ['a', 'b', 'c', 'c', 'c', 'b', 'a', 'c', 'a', 'b', 'c', 'b', 'a']

```

Создать словарь, в котором ключи – это содержимое списка B, а значения для ключей словаря – это сумма всех элементов списка A в соответствии с буквой, содержащийся на той же позиции в списке B. Пример результата программы: `{'a': 10, 'b': 15, 'c': 6}`.

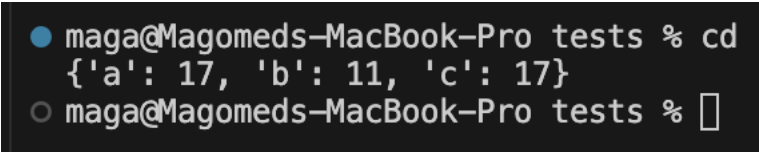
Рисунок 1.9 — Задание 5

Код задания представлен в листинге 1.5.

Листинг 1.5 — Код Задания 5

```
A = [1, 2, 3, 4, 2, 1, 3, 4, 5, 6, 5, 4, 3, 2]
B = ['a', 'b', 'c', 'c', 'c', 'b', 'a', 'c', 'a', 'a', 'b', 'c', 'b', 'a']
result = {}
for i in range(len(A)):
    updated = {B[i]: result[B[i]] + A[i] if B[i] in result else A[i]}
    result.update(updated)
print(result)
```

Результат выполнения Задания 5 представлен на рисунке 1.10

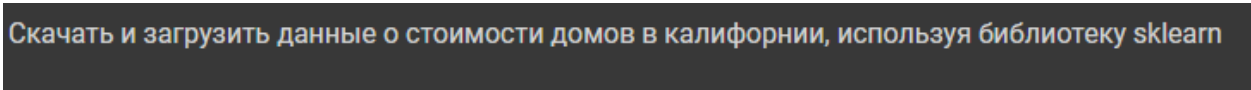


```
● maga@Magomeds-MacBook-Pro tests % cd
  { 'a': 17, 'b': 11, 'c': 17 }
○ maga@Magomeds-MacBook-Pro tests % █
```

Рисунок 1.10 — Результат выполнения Задания 5

1.6 Задание 6

Задание 6 представлено на рисунке 1.11.



Скачать и загрузить данные о стоимости домов в калифорнии, используя библиотеку sklearn

Рисунок 1.11 — Задание 6

Так как загруженный датасет является `sklearn.utils._bunch.Bunch`, а дальнейшие задания для `pandas.DataFrame`, необходимо перевести скачанный датасет в нужный формат.

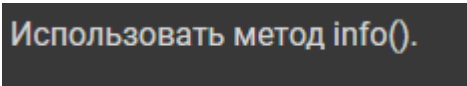
Код задания представлен в листинге 1.6.

Листинг 1.6 — Код Задания 6

```
from sklearn.datasets import fetch_california_housing
import pandas as pd
import numpy as np
data=fetch_california_housing(as_frame=True)
data1 = pd.DataFrame(data= np.c_[data['data'], data['target']], columns= data['feature_names'] + ['target'])
```

1.7 Задание 7

Задание 7 представлено на рисунке 1.12.



Использовать метод `info()`.

Рисунок 1.12 — Задание 7

Код задания представлен в листинге 1.7.

Листинг 1.7 — Код Задания 7

```
data1.info()
```

Результат выполнения Задания 7 представлен на рисунке 1.13

```
maga@Magomeds-MacBook-Pro tests % cd "/Use
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  -
0   MedInc      20640 non-null  float64
1   HouseAge    20640 non-null  float64
2   AveRooms    20640 non-null  float64
3   AveBedrms   20640 non-null  float64
4   Population  20640 non-null  float64
5   AveOccup    20640 non-null  float64
6   Latitude    20640 non-null  float64
7   Longitude   20640 non-null  float64
8   target      20640 non-null  float64
dtypes: float64(9)
memory usage: 1.4 MB
maga@Magomeds-MacBook-Pro tests %
```

Рисунок 1.13 — Результат выполнения Задания 7

1.8 Задание 8

Задание 8 представлено на рисунке 1.14.

Узнать, есть ли пропущенные значения, используя `isna().sum()`.

Рисунок 1.14 — Задание 8

Код задания представлен в листинге 1.8.

Листинг 1.8 — Код Задания 8

```
print(data1.isna().sum())
```

Результат выполнения Задания 8 представлен на рисунке 1.15

```

maga@Magomeds-MacB
MedInc          0
HouseAge        0
AveRooms        0
AveBedrms       0
Population      0
AveOccup        0
Latitude        0
Longitude       0
target         0
dtype: int64
maga@Magomeds-MacB

```

Рисунок 1.15 — Результат выполнения Задания 8

1.9 Задание 9

Задание 9 представлено на рисунке 1.16.

Вывести записи, где средний возраст домов в районе более 50 лет и население более 2500 человек, используя метод loc().

Рисунок 1.16 — Задание 9

Код задания представлен в листинге 1.9.

Листинг 1.9— Код Задания 9

```
print(data1.loc[(data1['HouseAge']>50) & (data1['Population']>2500)])
```

Результат выполнения Задания 9 представлен на рисунке 1.17

```

maga@Magomeds-MacBook-Pro tests % cd "/Users/maga/Desktop/tests/" && python3 main.py
MedInc HouseAge AveRooms AveBedrms Population AveOccup Latitude Longitude target
460      1.4012      52.0   3.105714  1.060000   3337.0    9.534286   37.87   -122.26  1.75000
4131     3.5349      52.0   4.646119  1.047945   2589.0    5.910959   34.13   -118.20  1.93600
4440     2.6806      52.0   4.806283  1.057592   3062.0    4.007853   34.08   -118.21  1.53000
5986     1.8750      52.0   4.500000  1.206349   2688.0    21.333333   34.10   -117.71  2.12500
7369     3.1901      52.0   4.730942  1.017937   3731.0    4.182735   33.97   -118.21  1.67600
8227     2.3305      52.0   3.488860  1.170380   3018.0    3.955439   33.78   -118.20  1.62500
13034    6.1359      52.0   8.275862  1.517241   6675.0    230.172414  38.69   -121.15  2.25000
15634    1.8295      52.0   2.628169  1.053521   2957.0    4.164789   37.80   -122.41  2.43800
15652    0.9000      52.0   2.237474  1.053535   3260.0    2.237474   37.80   -122.41  5.00001
15657    2.5166      52.0   2.839075  1.184049   3436.0    1.621520   37.79   -122.41  2.75000
15659    1.7240      52.0   2.278566  1.082348   4518.0    1.780142   37.79   -122.41  2.25000
15795    2.5755      52.0   3.402576  1.058776   2619.0    2.108696   37.77   -122.42  3.25000
15868    2.8135      52.0   4.584329  1.041169   2987.0    3.966799   37.76   -122.41  2.60300
maga@Magomeds-MacBook-Pro tests %

```

Рисунок 1.17 — Результат выполнения Задания 9

1.10 Задание 10

Задание 10 представлено на рисунке 1.18.

Узнать максимальное и минимальное значения медианной стоимости дома.

Рисунок 1.18 — Задание 10

Код задания представлен в листинге 1.10.

Листинг 1.10— Код Задания 10

```
print(data1.MedInc.max())  
print(data1.MedInc.min())
```

Результат выполнения Задания 10 представлен на рисунке 1.19

```
● maga@Magomeds  
  15.0001  
  0.4999  
○ maga@Magomeds
```

Рисунок 1.19 — Результат выполнения Задания 10

1.11 Задание 11

Задание 11 представлено на рисунке 1.20.

Используя метод `apply()`, вывести на экран название признака и его среднее значение.

Рисунок 1.20 — Задание 11

Код задания представлен в листинге 1.11.

Листинг 1.11 — Код Задания 11

```
def print_mean(column):  
    mean_value = column.mean()  
    print(f'Name: {column.name}, AvgValue: {mean_value}')  
  
data1.apply(print_mean, axis=0)
```

Результат выполнения Задания 11 представлен на рисунке 1.21

```
● maga@Magomeds-MacBook-Pro tests % cd "/Users/ma  
Name: MedInc, AvgValue: 3.8706710029069766  
Name: HouseAge, AvgValue: 28.639486434108527  
Name: AveRooms, AvgValue: 5.428999742190376  
Name: AveBedrms, AvgValue: 1.096675149606208  
Name: Population, AvgValue: 1425.4767441860465  
Name: AveOccup, AvgValue: 3.0706551594363742  
Name: Latitude, AvgValue: 35.63186143410853  
Name: Longitude, AvgValue: -119.56970445736432  
Name: target, AvgValue: 2.068558169089147  
○ maga@Magomeds-MacBook-Pro tests % █
```

Рисунок 1.21 — Результат выполнения Задания 11

1.12 Задание 12

Задание 12 представлено на рисунке 1.22.

```
morze = {'a': '.-', 'b': '-..', 'c': '-.-.', 'd': '-..-',  
        'e': '.', 'f': '..-.', 'g': '--.', 'h': '....',  
        'i': '..', 'j': '.---', 'k': '-.-', 'l': '-.-..',  
        'm': '--', 'n': '-.', 'o': '---', 'p': '--.-',  
        'q': '-.-.-', 'r': '.-.', 's': '...', 't': '-',  
        'u': '..-', 'v': '...-', 'w': '---', 'x': '---.',  
        'y': '---..', 'z': '----'}
```

На выходе: Выведите каждое слово исходного текста, закодированное азбукой Морзе. Количество строк в ответе должно совпадать с количеством слов в исходном тексте. Между закодированными буквами ставится ровно один пробел. Например, слово «Help» превратится в «.... .-.. -.-».

Перевод ..

Код задания представлен в листинге 1.12.

```
morze = {'a': '...', 'b': '...', 'c': '...', 'd': '...',
         'e': '...', 'f': '...', 'g': '...', 'h': '...',
         'i': '...', 'j': '...', 'k': '...', 'l': '...',
         'm': '...', 'n': '...', 'o': '...', 'p': '...',
         'q': '...', 'r': '...', 's': '...', 't': '...',
         'u': '...', 'v': '...', 'w': '...', 'x': '...',
         'y': '...', 'z': '...'}
```

```
text=input("Enter text: ")
text=text.lower()
for i in text:
    if i==" ":
        print("\n", end=" ")
    else:
        print(morzel[i], end=" ")
```

```
maga@Magomeds-MacBook-Pro tests % cd /
Enter text: privet ya perevel v morze
.-.-. .-. . . ...- . -
-.-. -
.-.-. . . .-. . ...- . .-. .
...-
-.- -.-. .-. -.-. . . %
maga@Magomeds-MacBook-Pro tests %
```

Рисунок 1.23 — Результат выполнения Задания 12