Надирян Г.О.

КИТ-36

Лабораторна робота №1

Тема: Основи мови Python

Мета: Розгляд середовища розробки, типів даних, операторів, функцій виведення і введення даних

Індивідуальні завдання

1. Ознайомитися з інтерактивним і сценарним режимами роботи середовища розробки програм на мові Python (IDLE).

2. Розробити програму на мові Python, яка виконує наступне:

за допомогою вбудованої системи допомоги (help>) отримує інформацію про вбудованої функції або операції, зазначеної в колонці "Help" табл. 5 (якщо обсяг даних великий - привести фрагмент отриманих даних):

1 - +;

2 - -;

3 - /;

4 - //;

5 -%;

*6 - **;*

7 - abs ();

8 - divmod ();

9 - pow ();

10 - round ();

11 - &;

12 - | ;

13 - ^;

14 - ~;

виконує введення двох аргументів - перший є цілим десятковим числом, а другий задається колонкою "Аргумент" табл. 5:

1 - вісімкове число;

2 - шестнадцатиричное число;

3 - число з плаваючою точкою;

після введення двох аргументів - виконує над ними арифметичну операцію, зазначену колонкою "Операція" / "арифметична" табл. 5;

перевіряє тип отриманого результату. Якщо він має значення float - перетворити результат в ціле число;

Вивести результат операції в вікно середовища розробки в системі числення за основою, зазначеному колонкою "Підстава" табл. 5:

виконує введення двох аргументів - кожен представляє собою ціле число, задане в двійковій системі і має довжину 8 бітів;

здійснює над цими аргументами побітову операцію, зазначену колонкою "Операція" / "побітова" табл. 5, виконану за допомогою відповідного спеціального методу;

вивести значення операндів і результат побітової операції в вікно середовища.

Таблица 5 – Перечень индивидуальных заданий

Номер п/п	Help	Аргумент	Основание	Операция	
				арифметическая	побитовая
12	12	3	2	//	<<

Результат

За допомогою вбудованої системи допомоги (help>) отримує інформацію про вбудовану операцію ‘|’ (Рисунок 1)

```
==================== RESTART: D:/Univer/!PYTHON/Labs/1.py ====================
Operator precedence
*******************

The following table summarizes the operator precedence in Python, from
lowest precedence (least binding) to highest precedence (most
binding).  Operators in the same box have the same precedence.  Unless
the syntax is explicitly given, operators are binary.  Operators in
the same box group left to right (except for exponentiation, which
groups from right to left).

Note that comparisons, membership tests, and identity tests, all have
the same precedence and have a left-to-right chaining feature as
described in the Comparisons section.

+-----------------------------------------------+---------------------------------------+
| Operator                                      | Description                           |
+===============================================+=======================================+
| "lambda"                                      | Lambda expression                     |
+-----------------------------------------------+---------------------------------------+
| "if" -- "else"                                | Conditional expression                |
+-----------------------------------------------+---------------------------------------+
| "or"                                          | Boolean OR                            |
+-----------------------------------------------+---------------------------------------+
```

Рисунок 1

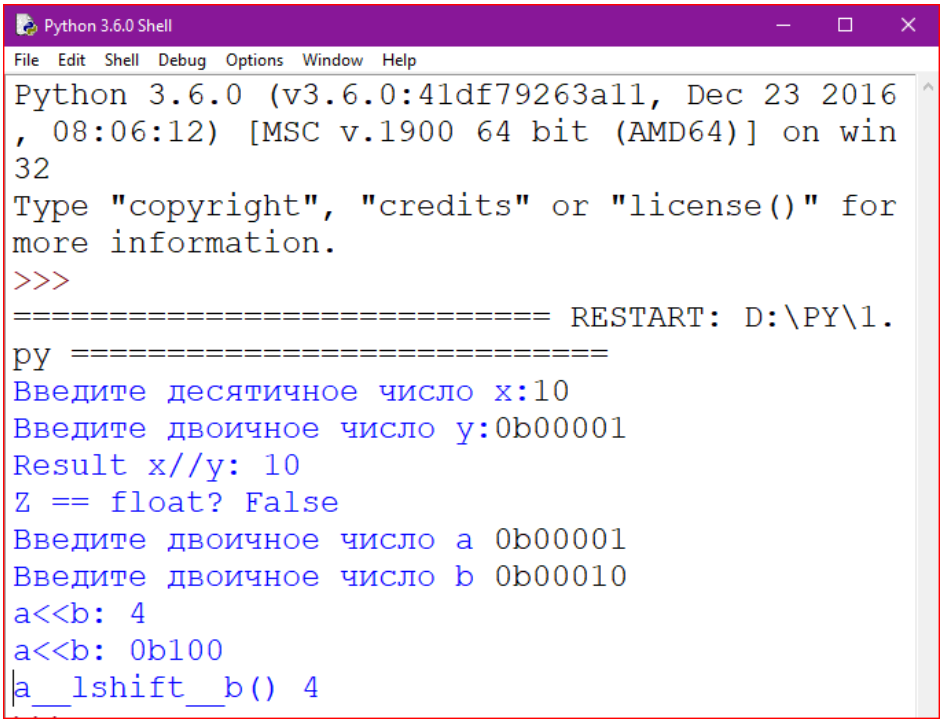

Рисунок 2

Код програми

```
output = help("|")
x = int ( input ( "Введите десятичное число x:" ),10)
y = int ( input ( "Введите двоичное число y:" ),2)
z = x//y;
print("Result x//y:",z);
test = isinstance (z, float)
print("Z == float?",test);
if test == True:
 isinstance (int (z),int)
#print(bin(z));
a = int ( input ( "Введите двоичное число a " ),2)
b = int ( input ( "Введите двоичное число a " ),2)
print("a<<b:",a<<b);
print("a<<b:",bin(a<<b));
print("a__lshift__b()",(a.__lshift__(b)))
```

Висновок

Розглянули середовище розробки, типів даних, операторі, функцій виведення і введення даних. Ознайомитися з інтерактивним і сценарним режимами роботи середовища розробки програм на мові Python

Лабораторна робота №2

Тема: Рядки

Мета: Розгляд способів створення тексту і функцій і методів роботи з ним

ІНДИВІДУАЛЬНІ ЗАВДАННЯ

Розробити програму на мові Python, яка виконує наступне:
Вводить прізвище, ім'я та по батькові студента у вигляді одного рядка S1.
Визначає зріз рядки S1 згідно колонці "Зріз" табл. 2.
Вводить шифр групи студента вигляді одного рядка S2.
Виконує конкатенацію рядків S1 і S2, формуючи рядок S3.
Застосовує до рядка S3 методи згідно колонці "Методи" табл. 2.
Перелік методів:
1. center ();
2. count ();
3. endwith ();
4. isalnum ();
5. isalpha ();
6. islower ();
7. ljust ();
8. replace ();
9. rindex ();
10. rfind ();
11. rjust ();
12. swapcase ().
Виводить на екран 5 питань, по одному на кожну тему згідно колонці "Теми" табл. 2.
Перелік тем:
1. Операції з числами;
2. Функції рядків;
3. Логічний тип;
4. Числа;
5. Завдання рядків;
6. Зрізи рядків;
7. Методи рядків;
8. Оператори введення / виведення;
9. Умовний оператор;
10. Оператори циклу.

Таблица 2 – Перечень индивидуальных заданий			
Номер п/п	**Срез**	**Методы**	**Темы**
12	3::2	1,5,9,11	6,7,3,4,5

Результат роботи

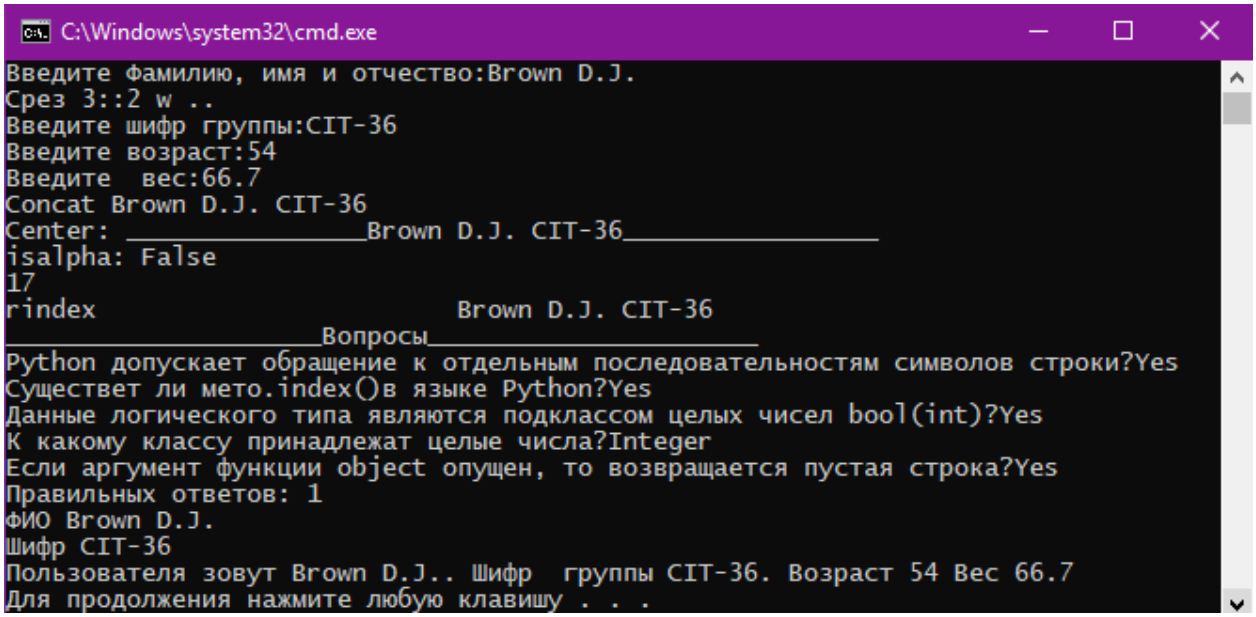

Рисунок 1 – Результат роботи

Код програми

```
S1 = ( input ( "Введите Фамилию, имя и отчество:",))
print("Срез 3::2",S1[3::2]);
S2 = ( input ( "Введите шифр группы:", ))
S3=S1+" "+S2;
age = int ( input ( "Введите возраст:" ))
weight = float(input("Введите вес:"))
print('Concat',S3);
print("Center:",S3.center(50,'_'));
print("isalpha:",S3.isalpha());
print(S3.rindex('));
print("rindex",S3.rjust(40));
print("Вопросы".center(50,'_'));
my_list=["Python допускает обращение к отдельным последовательностям символов строки?",
         "Существет ли мето.index()в языке Python?",
         "Данные логического типа являются подклассом целых чисел bool(int)?",
         "К какому классу принадлежат целые числа?",
         "Если аргумент функции object опущен, то возвращается пустая строка?"];
```

```
my_listKey = ["Да","Да","Да", "Integer","Да"]
count=0;
a0 = (input(my_list[0],))
if a0==my_listKey[0]:
    count+=1;
a1 = (input(my_list[1],))
if a1==my_listKey[1]:
    count+=1;
a2 = (input(my_list[2],))
if a2==my_listKey[2]:
    count+=1;

a3 = (input(my_list[3],))
if a3==my_listKey[3]:
    count+=1;

a4 = (input(my_list[4],))
if a4==my_listKey[4]:
    count+=1;
print("Правильных ответов:",count)
print("ФИО",S1);
print("Шифр",S2);
print(f"Пользователя зовут {S1}. Шифр  группы {S2}. Возраст {age} Вес {weight}")
```

Висновок

Розглянули способі створення тексту і функцій і методів роботи з ними. Розробили програму на мові Python, яка виконує наступне: Вводить прізвище, ім'я та по батькові студента у вигляді одного рядка S1,Визначає зріз рядки S1 Вводить шифр групи студента вигляді одного рядка S2. Виконує конкатенацію рядків S1 і S2, формуючи рядок S3. Застосовує до рядка S3 методи згідно колонці "Методи" табл. 2.

Лабораторна робота №3

Тема: Підключення модулів. ABC-класи. списки

Мета: Розгляд особливостей використання модулів, ABC-класів і списків

ІНДИВІДУАЛЬНІ ЗАВДАННЯ

Розробити програму на мові Python, яка виконує наступне:
1 Створює список a_list, елементами якого є об'єкти типів: int, float, bool, strl і list (елементи списку задаються в будь-якому порядку).
2 Для кожного елемента списку a_list визначає:
2.1 тип елемента;
2.2 ABC-клас (якщо елемент належить до деякого ABC-класу).
3 Для списку a_list виконує операції, номера яких задані в колонці "Операції" табл. №2:
2 - s [i: j] = t;
5 - s.insert (i, x);
8 - s.reverce ().
4 Створює рядок s, що містить слова або числа (див. Колонку "Рядок" / "Вміст" табл. №2), розділені символами, зазначеними в колонці "Рядок" / "Роздільник" табл. №2:
2 - дефіс ("-");
5 Перетворює рядок s в список b_list, елементами якого є рядки або числа (згідно колонки "Рядок" / "Вміст" табл. №2).
6 Виконує над елементами списку b_list функції, зазначені в колонці "Функції":
3 - min (), max ().
7 Виводить результати виконання функцій на екран.

Таблица 2 – Перечень индивидуальных заданий

Номер п/п	Операции	Строка		Фукции
		Содержимое	Разделитель	
12	5,2,8	числа	2	3

Робота програми

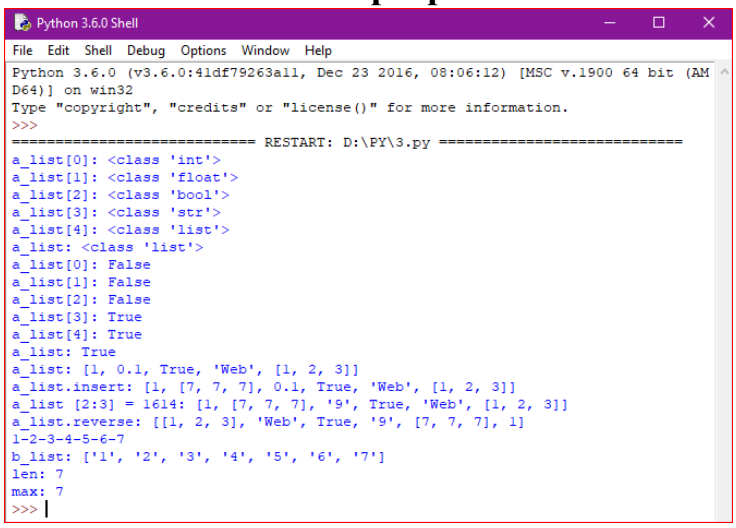

Рисунок 1

Код програми

```
a_list=[1,0.1,True, 'Web' , [1,2,3]];
print("a_list[0]:",type(a_list[0]));
print("a_list[1]:",type(a_list[1]));
print("a_list[2]:",type(a_list[2]));
print("a_list[3]:",type(a_list[3]));
print("a_list[4]:",type(a_list[4]));
print("a_list:",type(a_list));
from collections.abc import Container
print("a_list[0]:",isinstance ( a_list[0] , Container))
print("a_list[1]:",isinstance ( a_list[1] , Container))
print("a_list[2]:",isinstance ( a_list[2] , Container))
print("a_list[3]:",isinstance ( a_list[3] , Container))
print("a_list[4]:",isinstance ( a_list[4] , Container))
print("a_list:",isinstance ( a_list , Container))
print("a_list:",a_list);
a_list.insert(1,[7,7,7])
print("a_list.insert:" ,a_list);
a_list[2:3] = '9'
print("a_list [2:3] = 1614:", a_list);
a_list.reverse()
print("a_list.reverse:",a_list);
str_s = ['1','2','3','4','5','6','7'];
sep = "-";
#'-'.join(['1','2','3','4','5','6']);
print('-'.join(str_s));
b_list = list(str_s);
print("b_list:",b_list);
print("len:",len(b_list));
print("max:",max(b_list));
```

Висновок

Розгляунли особливості використання модулів, ABC-класів і списків

Лабораторна робота №4

Тема: Кортежі. Діапазони. двійкові послідовності

Мета: Вивчення способів роботи з кортежами, діапазонами і двійковими послідовностями

ІНДИВІДУАЛЬНІ ЗАВДАННЯ

1 Розробити програму на мові Python, яка виконує наступне:

1.1 Створює кортеж a_tuple, елементами якого є записи наступних типів (див. Колонку "Тип" табл. №2):

- *1 - ціле число;*
- *2 - число з плаваючою точкою;*
- *3 - логічне значення;*
- *4 - рядок;*
- *5 - список;*
- *6 - кортеж.*

1.2 Перевіряє, чи є об'єкт a_tuple:

- *контейнером;*
- *ітератором;*
- *ітерабельним об'єктом;*
- *послідовністю;*
- *змінюваною послідовністю.*

1.3 Змінити значення одного з елементів об'єкта a_tuple шляхом перетворення його спочатку в список, а потім знову в кортеж.

1.4 Створює об'єкт a_range згідно колонці "Діапазон" табл. №2.

1.5 Виводить на екран значення об'єкта a_range, використовуючи (див. Колонку "Висновок" табл. №2):

- *1 - оператор for;*
- *2 - конструктор списку;*
- *3 - конструктор кортежу.*
- *2. Розробити програму, яка:*

2.1 Виконує всі функції програми "ana.py" (див. Підрозділ 1.3). При цьому формує до п'яти анаграм середньої складності (ступінь складності визначає студент).

- 2.2 Розширює функціональність програми в такий спосіб (див. Колонку "Варіант" т*абл. №2):*
- *Варіант 1 - якщо користувач дав правильне рішення, то йому надається можливість вибрати більш складний варіант анаграми (всього три рівня складності);*
- *Варіант 2 - якщо користувач дав неправильне рішення, то йому надається можливість вибрати більш легкий варіант анаграми (всього три рівня складності);*
- *Варіант 3 - користувачеві надається можливість отримати підказку: першу букву слова, потім Слуда за нею і т.д. (Всього три підказки);*
- *Варіант 4 - користувачеві надається можливість отримати підказку: останню букву слова, потім попередню і т.д. (Всього три підказки);*
- *Варіант 5 - користувачеві надається можливість отримати наступну підказку: на його запит у вигляді рядка, наприклад, *** a **, що означає - чи містить вихідне слово букву "a" в зазначеному місці слова, чи ні. Програма відповідає ствердно, якщо це так, і негативно в іншому випадку (всього три підказки).*

Таблица 2 – Перечень индивидуальных заданий

Номер п/п	Тип элемента	Диапазон	Вывод	Вариант
12	2,3,5,6	11	3	2

Робота програми 4_1.py

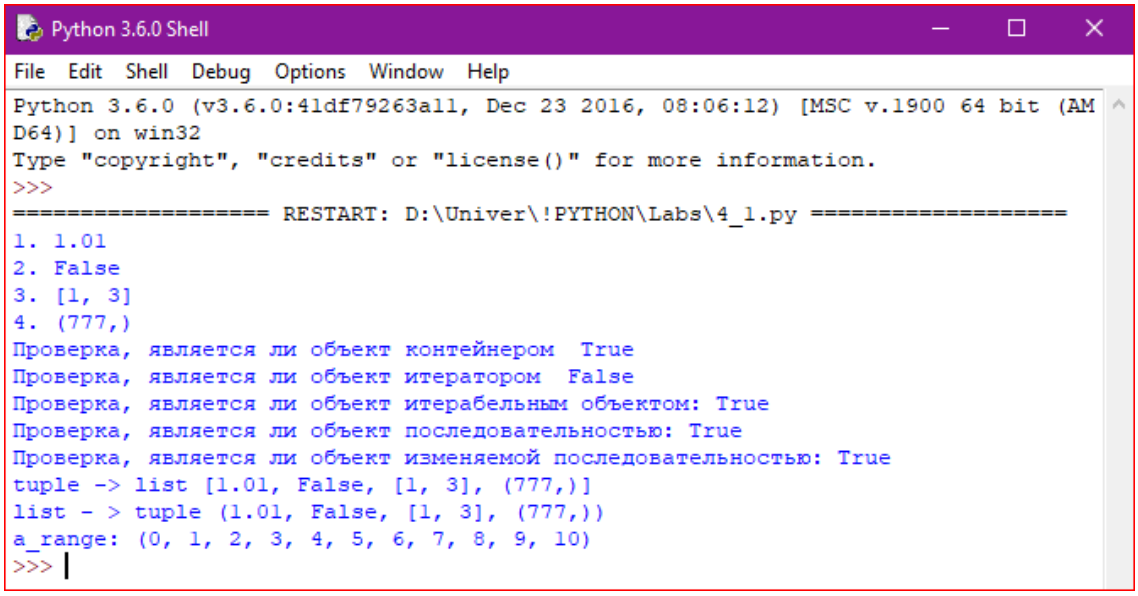

Рисунок 1

Робота програми 4_2anagramma.py

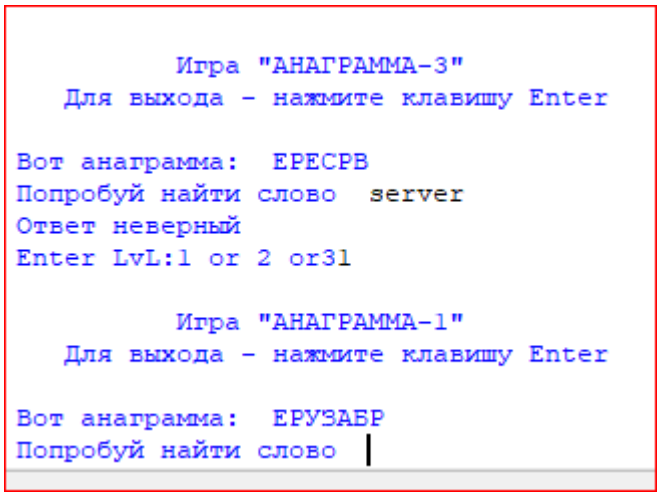

Рисунок 2

Код програми 4_1.py

```
a_tuple=[ 1.01 , False , [1,3],(777,) ]
for i in range (1, len (a_tuple)+1):
    print (i, a_tuple[i-1], sep= '. ' )
from collections import Sequence, MutableSequence,Iterable,Iterator,Container
print("Проверка, является ли объект контейнером ",isinstance ( a_tuple , Container))
print("Проверка, является ли объект итератором ",isinstance ( a_tuple , Iterator))
print("Проверка, является ли объект итерабельным объектом:",isinstance ( a_tuple , Iterable))
print("Проверка, является ли объект последовательностью:",isinstance(a_tuple, Sequence))
print("Проверка, является ли объект изменяемой последовательностью:",isinstance (a_tuple, MutableSequence))
print("tuple -> list",list(a_tuple))
print("list - > tuple",tuple(a_tuple))
a_range = range(11);
print("a_range:",tuple(a_range));
```

Код програми 4_2anagramma.py

```
# Создание анаграмм
WORDS=('компьютер',
'питон' ,
'сервер' ,
'клиент' ,
'браузер' ,
'протокол' ,
```

```
'программа' ,
'процессор' ,
'контекст' )
from random import seed,choice,randrange
enter =None
count=0;
while count<5:
    count=count+1;
    seed()
    word=choice(WORDS)
    correct=word
    ana=''
    while word:
        pos=randrange(len(word))
        print(pos)
        ana+=word[pos]
        word=word[:pos]+word[pos+1:]
        print(word)
    #word=word[:pos]+word[-1:pos:-1]
    #word=word[-1:pos:-1]+word[:pos]
        print(word);
    print('''
          Игра "АНАГРАММА-3"
   Для выхода - нажмите клавишу Enter''')
    print('\nВот анаграмма: ', ana.upper())
    ans=input('Попробуй найти слово  ').casefold()

    if ans !=correct and ans !='':
        print('Ответ неверный')
        #print("Выберите уровень:\t1,\t2,\t3");
        enter = input('Enter LvL:1 or 2 or3');
        if enter=='1':
            word=choice(WORDS)
            correct=word
            ana=''
            while word:
                pos=randrange(len(word))
                #print(pos)
                ana+=word[pos]
                #word=word[:pos]+word[pos+1:]
                #print(word)
                word=word[:pos]+word[-1:pos:-1]
```

```
            #word=word[-1:pos:-1]+word[:pos]
            #print(word);
        print('''
      Игра "АНАГРАММА-1"
Для выхода - нажмите клавишу Enter''')
        print('\nВот анаграмма: ', ana.upper())
        ans=input('Попробуй найти слово  ').casefold()
    elif enter=='2':
        word=choice(WORDS)
        correct=word
        ana=''
        while word:
            pos=randrange(len(word))
            #print(pos)
            ana+=word[pos]
            #word=word[:pos]+word[pos+1:]
            #print(word)
            #word=word[:pos]+word[-1:pos:-1]
            word=word[-1:pos:-1]+word[:pos]
            #print(word);
        print('''
      Игра "АНАГРАММА-2"
Для выхода - нажмите клавишу Enter''')
        print('\nВот анаграмма: ', ana.upper())
        ans=input('Попробуй найти слово  ').casefold()

    else:
 #while ans !=correct and ans !='':
  #ans=input('Попробуй еще раз ').casefold()
        if ans==correct:
            print('Молодец!')
            print('Спасибо за игру')
            input()
```

Висновок

Вивчили способи роботи з кортежами, діапазонами і двійковими послідовностями

Лабораторна робота №5

Тема: Сила-силенна. Словники. Генератори

Мета: Розгляд способів роботи з множинами, словниками та генераторами

Индивидуальные задания

1.Розроблена програму на мові Python, яка виконує наступне:

2.Создать безліч a_set, що містить не менше 7 елементів будь-яких дозволених типів, за допомогою:

 литерала.

Змінює безліч a_set за допомогою методів:

 Update ()

 Pop ()

3.Создает ітерабельний об'єкт it_ob, що містить не менше трьох елементів, наявних в об'єкті a_set, і перевірити, чи всі елементи it_ob хешіруеми. Якщо немає - замінити нехешіруемие елементи хешіруемимі.

4.Преобразует об'єкт it_ob в безліч b_set і виконує над множинами a_set і b_set операції:

 union ();

5.Создает словник a_dict за допомогою:

 конструктора з іменованнимі аргументами;

Код програми

```
from collections.abc import Hashable, Iterable, Collection, Sequence, Set, MutableSet,MutableSequence
a_set = {1, 1.2, (4,5), 'ABC', True,False,26,25}
print("a_set:",a_set)
a_set.update([1614,2],{'b','c'})
print("a_set.update:",a_set)
a_set.pop()
print("a_set.pop:",a_set)
it_ob = ([1,3,4,7])
print("Iterable it_ob:",isinstance(it_ob,Iterable))
b_set = set(it_ob);
a_set = a_set.union(b_set)
print("a_set.union(b_set):",a_set)
a_dict= dict ({'AngelsDemons':1,'TheDaVinciCode':2,'TheLostSymbol':3})
print("a_dict",a_dict)
print("a_dict.keys()",a_dict.keys())
print("a_dict.pop count element:",a_dict.pop('TheDaVinciCode'))
print ("a_dict.values():",a_dict.values())
```

Робота программа

```
========== RESTART: D:\Univer\!PYTHON\Labs\5.py ==========
a_set: {False, 1, 1.2, 26, (4, 5), 25, 'ABC'}
a_set.update: {False, 1, 1.2, 26, 2, (4, 5), 'c', 1614, 'b', 25, 'ABC'}
a_set.pop: {1, 1.2, 26, 2, (4, 5), 'c', 1614, 'b', 25, 'ABC'}
Iterable it_ob: True
a_set.union(b_set): {1, 1.2, 2, 3, 4, 7, (4, 5), 'c', 1614, 25, 26, 'ABC', 'b'}
a_dict {'AngelsDemons': 1, 'TheDaVinciCode': 2, 'TheLostSymbol': 3}
a_dict.keys() dict_keys(['AngelsDemons', 'TheDaVinciCode', 'TheLostSymbol'])
a_dict.pop count element: 2
a_dict.values(): dict_values([1, 3])
```

Рисунок 1

Висновок

Розглянули способі роботи з множинами, словниками та генераторами.

Лабораторна робота №6

Тема: Розробка функцій і модулів користувача

Мета: Розгляд особливостей розробки функцій і модулів користувача

індивідуальні завдання

Розробити програму на мові Python, в якій:

1 Визначено і виконана функція func1 () з аргументами у вигляді списку чисел (цілих і з плаваючою точкою), яка виконує операцію, задану колонкою "Операція" табл. №1):

1 - визначення суми квадратів елементів списку;

2 - визначення максимального числа серед елементів списку;

3 - визначення квадратних коренів елементів списку;

4 - визначення мінімального числа серед елементів списку;

5 - визначення середнього значення елементів списку;

6 - визначення різниці між сумою парних чисел і сумою непарних чисел (якщо число не ціле - привести до цілого);

7 - визначення різниці між сумою кубів і сумою квадратів елементів списку.

2.1 Створено словник a_dict (числом елементів не менше 8), ключі якого іменуються довільно, а значення задані у вигляді, зазначеному колонкою "Вид значень" табл. №1):

1 - латинські літери;

2 - цифри;

3 - літери кирилиці;

4 - найменування вбудованих функцій;

5 - найменування операторів.

При цьому окремі ключі (числом не менше трьох) повинні мати однакове значення.

2.2 Визначено функція func2 (), яка має два аргументи, перший - у вигляді словника, другий - вказує значення ключа словника. Функція func2 () повертає

список ключів словника, значення яких збігаються зі значеннями другого аргументу.

2.3 Перевірено робота функції func2 (), при виклику якої в якості першого аргументу задане словник a_dict, а в якості другого аргументу - значення, яке мають кілька ключів словника.

3.1 Створено список a_list, елементи якого мають тип, вказаний колонкою "Тип" табл. №1):

1 - числа;

2 - логічні значення;

3 - рядки;

4 - списки з елементами у вигляді чисел;

5 - списки з елементами у вигляді рядків.

3.2 Визначено функція func3 (), яка перетворює кожен елемент заданого списку a_list в ціле число (механізм перетворення - на розсуд студента).

3.3 Виконана з використанням функції func3 () і методу sort () сортування елементів списку a_list:

для парних номерів індивідуального завдання - по зростанню;

для непарних номерів індивідуального завдання - по спадаючій.

4.1 Задана рядок str_code, що містить невеликий фрагмент коду на мові Python і отриманий скомпільований за допомогою вбудованої функції compile () код - comp_code.

4.2 За допомогою вбудованої функції exec () код comp_code виконаний.

5 Для програми і однією з функцій програми визначено безліч глобальних імен та безліч локальних імен;

6. Створити свій власний модуль (див. Підрозділ 3), що містить довільну функцію (або функції), записати цей модуль в бібліотеку Lib, завантажити його і викликати на виконання одну або кілька функцій модуля. (В-12)

Код програми

```
#Определена и выполнена функция func1() с аргументами в виде списка чисел (целых и с
плавающей точкой), которая определяет среднее значения элементов списка;
#Function AVG
def list_mean(nums):
    sumof = 0
    global num_of
    num_of = len(nums)
    mean = 0
    for i in nums:
        sumof = sumof + i
        mean = sumof / num_of
    return float(mean)

print ("AVG:",list_mean([1,2]))
a_dict = dict(typeBool='bool(x)', typeObject='type(object)',
typeFloat="float([X])",typeByteArray="bytearray(...)",typeComplex="complex()",typeOb="Object
",typeObj="Object",typeObjec="Object")
def list_of_keys (args,typeFunc):
    l_type=[]
    for k in args:
        if args[k]==typeFunc:
            l_type+=[k]
    return l_type
print(list_of_keys( a_dict,"Object"))
a_list = ['21','12','3','43','55','46','0']
def func3(str_list):
    int_list = [int(n) for n in str_list]
    int_list.sort()
    return int_list
print(func3(a_list))

def func4():
    program = '''
global ham
ham = "30.06.1998"
print("Python the best!!!","Global:",ham)
'''
    exec(program)
import sys
print(sys.exec_prefix)

import my_mod
my_modPrint  = my_mod.list_mean([9,11])
print(my_modPrint)
```

Робота програми

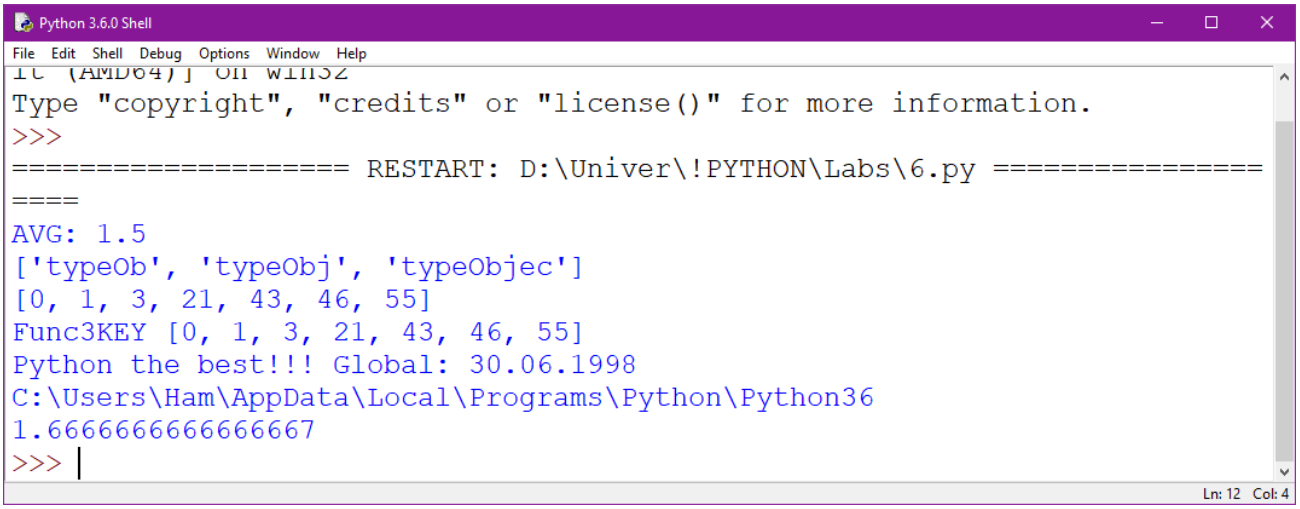

Рисунок 1 – Робота програми

Висновок

Розроботали функцій і модулі користувача. Розглянули особливості розробки функцій і модулів користувача.

Лабораторна робота №7

Тема: Розробка CGI скриптів на мові Python

Мета: Розгляд особливостей розробки серверних додатків

Індивідуальні завдання

Розробити програму на базі технології "клієнт-сервер". При цьому клієнтська частина передає на сервер ім'я пользователькля і номера функцій, а серверна частина, написана на мові Python, виконує ці функції і повертає користувачеві результат.

До завдань клієнтської частини входить: розробити HTML-документ, що містить форму, обов'язковими елементами якої є:

текстове поле для введення імені користувача;

теги (<select> або <input type = "checkbox">), задані колонкою "Теги" табл. 1, які дозволяють зробити користувачеві наступний вибір:

виконувати чи не виконувати функцію першої групи;

виконувати чи не виконувати функцію другої групи;

виконувати чи не виконувати функцію третьої групи;

кнопки для скидання даних форми і для передачі їх на сервер.

Функції по групах діляться наступним чином:

група 1:

а) - знайти суму позитивних чисел;

б) - знайти суму парних чисел;

в) - знайти твір чисел, узятих по абсолютній величині;

г) - знайти середнє аріфментіческое чисел;

д) - знайти твір непарних чисел.

група 2:

а) - знайти мінімальне число серед позитивних чисел;

б) - знайти максимальне число серед негативних чисел;

в) - знайти мінімальне число серед чисел, узятих по абсолютній величині.

група 3:

а) - виконати сортування чисел, узятих по абсолютній величині;

б) - виконати сортування чисел, узятих по молулю 3.

До завдань серверної частини входить:

розробка трьох функцій, заданих колонкою "Функції" табл. 1;

генерація випадкових чисел, їх діапазон і кількість задані колонками "Діапазон" і "Кількість" табл. 1);

прийом і аналіз даних, отриманих з форми з метою визначення:

імені користувача;
тих функцій, які треба виконувати;
виконання зазначених функцій над отриманими випадковими числами;
формування HTML-документа, в якому вказані випадкові числа і результати обчислень, і передача його користувачеві.

Код програми файлу index.html

```
<!DOCTYPE html>
<html>
<head>
<title>Лабораторна робота 7</title>
<meta charset="utf-8">
<link rel="stylesheet" href="lab.css">
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>
<h1 = align="center">Лабораторна робота 7</h1>
<form method="POST" action="/scripts/lab07.py">
        <table align="center" style="color:#000080; font-size:6.5mm">
                <tr> <td> Введите имя:
                <td> <input type="text" size="10" name="name">
                <tr> <td cols="2">
                <tr> <td> Найти сумму четных чисел ?
                <td> <input type="checkbox" name="check1">
                <tr> <td> Найти минимальное число среди чисел, взятых по абсолютной
величине ?
                <td> <input type="checkbox" name="check2">
                <tr> <td> Выполнить сортировку чисел, взятых по молулю 3 ?
                <td> <input type="checkbox" name="check3">
                <tr> <td> <input type="reset" value="Reset">   
                <input type="submit" value="Send" >
        </table>
</form>
</body>
</html>
```

Код програми файлу lab07.py

```
#!C:\Python3_7\python.exe
# -*- coding: utf-8 -*-
import cgi, cgitb
import random
from os import environ as env
print("Content-Type: text/html")
print()
```

```
def func1(*argslist):
    sum = 0
    for arg in argslist:
        if arg % 2 == 0:
            sum += arg
    return sum
def func2(*argslist):
    min = 100
    result = 0
    for arg in argslist:
        temp = abs(arg)
        if temp < min:
            min = temp
            result = arg
    return result
def func3(*argslist):
    res = []
    for arg in argslist:
        if arg % 3 == 0:
            res.append(arg)
        #else:
            #res.append(arg);
    res.sort()
    return res
H   =   "<p   style='text-align:center;color:blue;   font-size:7mm;   font-
weight:bold'>"
H2  =  "<p  style='text-align:center;color:green;  font-size:5.5mm;  font-
style:italic; font-weight:bold'>"
D = "<p style='margin-left:1.5cm;color:red; font-size:5.7mm'>"
# from my_mod import *
cgitb.enable(logdir="C:/Server/bin/Apache24/logs", context=7)
form = cgi.parse(keep_blank_values=1) #Этот метод возвращает данные HTML-
формы, полученные сервером, в виде словаря form,
name = 'клієнт'
if form['name'][0]:#доступ к значениям полей HTML-формы можно получить
следующим образом:
    name = form['name'][0]
print(H2, "Ни хау, ", name, "!", sep="")
if "check1" in form or "check2" in form or "check3" in form:
    print('Вам надаються:')
result = []
for x in range(11):
    result.append(random.randint(-55,66));
print(D, result);
if 'check1' in form:
    print(H, "Сумма четных чисел:")
    print(D, func1(*result))
if 'check2' in form:
```

```
    print(H, "Минимальное число среди чисел, взятых по абсолютной
величине:")
    print(D, func2(*result))
if 'check3' in form:
    print(H, "Выполнить сортировку чисел, взятых % 3.")
    print(D, func3(*result))
```

Робота програми

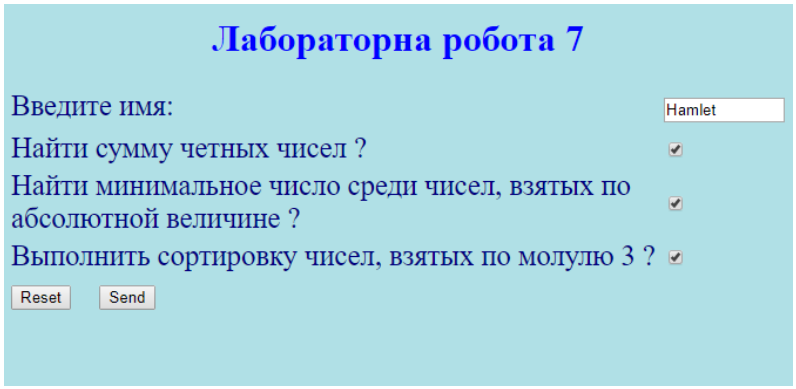

Рисунок 1 – Вигляд клієнтської частини

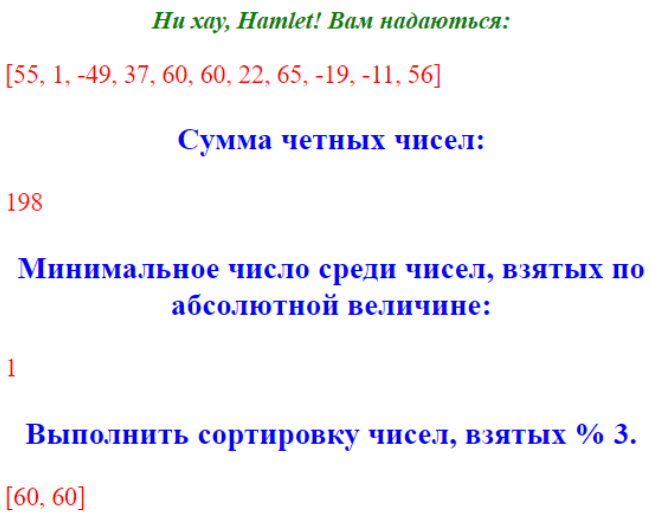

Рисунок 2 – Вигляд серверної частини

Висновок

Розробли CGI скрипті на мові Python. Розглянули особливості розробки серверних додатків Розробили програму на базі технології "клієнт-сервер". При цьому клієнтська частина передає на сервер ім'я користувача і номера функцій, а серверна частина, написана на мові Python, виконує ці функції і повертає користувачеві результат.

Лабораторна робота №8

Тема: Обробка винятків. Робота з файлами

Мета: Розгляд способів роботи з файлами

Індивідуальні завдання

Розробити програму на базі технології "клієнт-сервер", клієнтська частина якого повинна являти собою HTML-документ, що містить форму, обов'язковими елементами якої є:

три текстових поля для завдання даних, вибраних з нижче наведеного списку згідно колонці "Дані" табл. 2:

1 - Прізвище, ініціали;

2 - Найменування університету;

3 - Найменування факультету;

4 - Найменування кафедри;

5 - Найменування спеціальності;

6 - Курс;

7 - Найменування групи;

8 - Телефон;

9 - Мобільний телефон;

10 - Електронна пошта;

кнопка для передачі даних на сервер.

Серверна частина повинна виконати наступне:

прийняти дані з форми;

встановити поточний каталог і створити в ньому каталог "CGI";

створити в каталозі "CGI" відкриті для запису рядків файли "name.txt" і "values.txt";

записати прізвище та ініціали в файл "name.txt", інші отримані дані - в файл "values.txt", використовуючи методи, задані колонкою "Методи" / "Запис" табл. 2 (перша цифра - для файлу "name.txt", друга - для файлу "values.txt"):

1 - write ();

2 - writelines ();

визначити довжину файлів "name.txt" і "values.txt";

прочитати і передати клієнту вміст файлу "values.txt" за допомогою методу, заданого колонкою "Методи" / "Читання" табл. 2:

1 - read ();

2 - readline ();

3 - readlines ();

створити за допомогою вбудованих функцій range () і bytes () (див. розділи 2 і 3) лаб. раб. №4) послідовність чисел з довжиною, заданою колонкою "Довжина" табл. 2, і записати її в двійковий файл "binary_data.dat";

прочитати значення байта файлу "binary_data.dat", номер якого заданий колонкою "Номер" табл. 2;

перемістити покажчик файлу відповідно значенням колонок "Зсув" і "Звідки" табл. 2 і прочитати значення трьох байтів;

перевірити з використанням оператора try і вбудованої функції assert (), чи виконується умова, заданий колонкою "Умова" табл. 2, для цілого числа x, введеного функцією input ():

x є числом виду x2 + 3x + 7;

Код програми файлу index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Лабораторна робота №8</title>
<link rel="stylesheet" href="lab.css">
</head>
<body>
<h1>Лабораторна робота №8</h1>
<form action="/scripts/lab08.py">
<p>Фамилия, инициалы:<br/><input name="famInic"></p>
<p>Наименование кафедры:<br/><input name="spec"></p>
<p>Наименование группы:<br/><input name="group"></p>
<input type="submit">
```

```
</form>
</body>
</html>
```

Код програми файлу lab08.py

```
#!C:\Python3_7\python.exe
# -*- coding: utf-8 -*-
print("Content-Type: text/html")
print()
import cgi
import cgitb
import os
H = "<p style='text-align:center;color:#0000b0; font-size:7mm; font-weight:bold'>"
H2 = "<p style='text-align:center;color:#00b0b0; font-size:7.5mm; font-style:italic; font-weight:bold'>"
D = "<p style='margin-left:1.5cm;color:#0000b0; font-size:5.7mm'>"
cgitb.enable(logdir="C:/Server/bin/Apache24/logs", context=7)
form = cgi.parse(keep_blank_values=1)#Этот метод возвращает данные HTML-формы, полученные сервером, в виде словаря form,
f1 = open('names.txt', 'w')
f2 = open('values.txt', 'w')
for x in form.keys():
        f1.write(x + '\n')
        f2.writelines(form[x][0] + '\n')#доступ к значениям полей HTML-формы можно получить следующим образом:
f1.close() #переносит все сделанные в файле изменения на диск
#Метод close() также может быть использован для закрытия двоичных файло
f2.close()#переносит все сделанные в файле изменения на диск
print(H, "names.txt:" + str(os.path.getsize('names.txt')))
print(H, "values.txt:" + str(os.path.getsize('values.txt')))
f1 = open('values.txt', 'r')
str = ""
while True:
        line = f1.read()
        if not line: break
        else : str += line
f1.close()
print(D, "values.txt:" + str)
ss = bytes(range(9))
f1 = open('binary_data.dat', 'wb')
f1.write(ss)
f1.close()
f1 = open('binary_data.dat', 'rb')
f1.seek(8)
print(D, "binary_data.dat" + f1.read(1).__str__())
f1.seek(-5, 2)
print(D, "binary_data.dat  -5:" + f1.read(3).__str__())
f1.close()
```

Код програми файлу lab08_2.py

```
try:
```

```
    n = int(input("Введите число: "))
    sum1 = 0
    for i in range(1, n):
        if(n % i == 0):
            sum1 = sum1 + i
    #if (sum1 == n):
    assert sum1==n
    print("Число совершенное")
except AssertionError:print ('Число не совршенное')
```

Робота програми

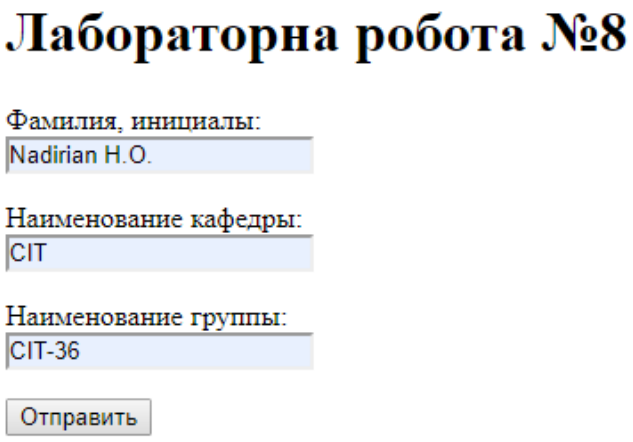

Рисунок 1 – Вигляд клієнтської частини

Рисунок 2 – Вигляд серверної частини

Висновок

Розробили програму на базі технології "клієнт-сервер", клієнтська частина которого винна являти собою HTML-документ, что містіть форму.

Лабораторна робота №9

Тема: Регулярні вирази

Мета: Вивчення засобів мови Python для роботи з регулярними виразами

Індивідуальні завдання

Розробити програму на базі технології "клієнт-сервер", клієнтська частина котра повинна являти собою форму, что містіть:
три текстових поля для запровадження
Даних користувача:
поле для запровадження факультету и групи (name = "data1");
поле для запровадження імені студента (name = "data2");
поле для запровадження номера залікової книжки (ЗК) и номера телефону (name = "data3");
текстове поле для запровадження літерала;
кнопка для передачі Даних на сервер.
До завдання серверної частині входити:
Прийняти дані з форми;
Сформувати три шаблони регулярних виразів для Перевірки даних "data1" "data2" і "data3" згідно колонкам "Данніе1" / "Шаблон", "Данніе2" / "Шаблон" і "Данніе3" / "Шаблон" табл. 1. дані, що вводяться мають постійну и змінну частині (остання вказана в фігурних дужках). Например, формат "Група: {}" означає, что має буті введено слово "Група:", а потім Назву групи (фігурні дужки не вводяться);
Сформувати шаблон РВ для Перевірки правильності завдання літерала згідно колонці "Літерал":
1 - літерала списку, что містить наступні елементи:
1-й елемент - рядок;
2-й елемент - число з плаваючих точкою;
3-й елемент - логічного типу;
2 - літерала кортежу; що містить наступні елементи:
1-й елемент - ціле число;
2-й елемент - список з трьох рядків;
3 - литерала діапазону;
4 - литерала безлічі, что містіть следующие елементи:
1-й елемент - кортеж з черірех ціліх чисел;
2-й елемент - рядок;
5 - литерала словника, что містіть два елементи:
ключ 1-го елемента - рядок, значення - рядок;
ключ 2-го елемента - ціле число, значення - число з плаваючих точкою;
використовувати для Перевірки Даних Функції або методи згідно "Данніе1" / "Метод", "Данніе2" / "Метод" та "Данніе3" / "Метод" табл. 1 (для Перевірки

колонки "Літерал" використовувати метод, Який НЕ використовувався при перевірках даних "data1" "data2" і "data3"):
1 - функцію match () модуля re;
2 - функцію search () модуля re;
3 - метод match () класу RegexpObject;
4 - метод search () класу RegexpObject;
за результатами всіх перевірок. Передати користувачу відповідне повідомлення.

Код html файлу index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Лабораторна робота 9 Надирян Г.О. </title>
<link rel="stylesheet" href="lab.css">
</head>
<body>
<h1>Регулярные выражения</h1>
<form action="/scripts/lab09.py">
<p>Факультет и группа:<br/><input
name="data1"></p>
<p>Имя студента:<br/><input name="data2"></p>
<p>Номер телефону:<br/><input name="data3">
</p>
<p>Литерал:<br/><input name="literal"></p>
<input type="submit">
</form>
</body>
</html>
```

Код програми lab09.py

```
#!C:\Python3_7\python.exe
# -*- coding: utf-8 -*-

print("Content-Type: text/html")

print()

import cgi

import cgitb

import re

H = "<p style='text-align:center;color:#0000b0; font-size:7mm; font-weight:bold'>"

H2 = "<p style='text-align:center;color:#00b0b0; font-size:7.5mm; font-style:italic; font-
weight:bold'>"

D = "<p style='margin-left:1.5cm;color:#0000b0; font-size:5.7mm'>"
```

```
cgitb.enable(logdir="C:\Server\bin\Apache24\logs", context=7)
form = cgi.parse(keep_blank_values=1)
regexUniver = re.compile("Faculty [A-Z a-z-]+\,st\.\:[A-Z a-z-]+\d+")
regexLastName = re.compile("N.Lastname [A-Z]\.[A-Z a-z]+")
# regexLiteral = re.compile("[([1-9]\d*["A-Z A-z"]+"["A-Z a-z"]+)]")
#\[[A-Z a-z]+\,[A-Z a-z]+\]
regexLiteral = re.compile("\([-+]?[0-9]+\,\[[\"A-Z a-z\"]+\,[\"A-Z a-z\"]+\]\)")
res1 = regexUniver.match(form["data1"][0])
res2 = regexLastName.match(form["data2"][0])
res3 = re.search("Number \d{6}", form["data3"][0])
res4 = regexLiteral.search(form["literal"][0])
if res1 :
   print(D, "Дані res1 введено правильно.")
if res2:
        print(D, "Дані res2 введено правильно.")
if res3:
        print(D, "Дані res3 введено правильно.")
if res4:
        print(D, "Дані res4 введено правильно.")
else:
   print(D, "Дані введено неправильно.")
```

Робота програми

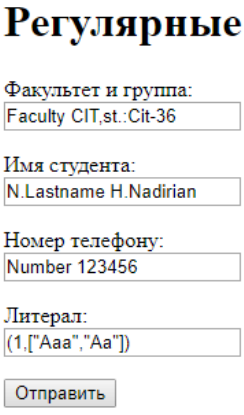

Рисунок 1 – Вигляд клієнтської частини

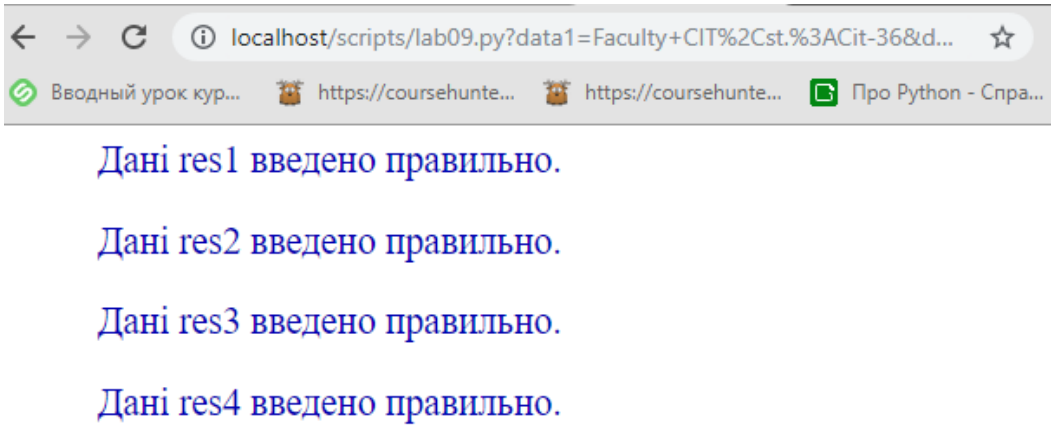

Рисунок 2 – Вигляд серверної частини

Вивчення засобів мови Python для роботи з регулярними виразами

Індивідуальні завдання

Розробили програму на базі технології "клієнт-сервер", клієнтська частина котра повинна являти собою форму, що містіть:

три текстових поля для запровадження

Лабораторна робота №10

Тема: Використання технології AJAX при розробці клієнт-серверного додатка

Мета: Розгляд особливостей роботи серверного додатка, що використовує технологію AJAX

Індивідуальні завдання

Розробити програму на базі технологій "клієнт-сервер" і AJAX, яка здійснює тестування користувача на стороні клієнта.

Клієнтську частину реалізувати в вигляді HTML-документа з включеними фрагментами JavaScript, завданнями якого є:

- прийом імені користувача (спосіб реалізації вказано в табл. 1);
- передача на сервер імені користувача з використанням AJAX-технології;
- прийом з сервера тестів з використанням AJAX-технології та візуалізація їх на веб-сторінці (спосіб реалізації вказано в табл. 1);
- прийом відповідей користувача (спосіб реалізації вказано в табл. 1);
- прийом з сервера результатів тестування з використанням AJAX-технології та візуалізація їх на веб-сторінці (так само, як і тести).

До завдань серверної частини програми входить:

- Розробка не менше чотирьох тестів у вигляді списку рядків по одній з наступних тем (мова Python):

  1) рядки;

  2) кортежі і списки;

  3) числа й сериализация;

  4) словники;

  5) оператори;

  6) функції і сортування елементів;

  7) робота з файлами;

  і збереження його на диску у вигляді виконуваного файлу. Номер теми для варіанту вказано в табл. 1;

- Кожен тест складається з заголовка, в якому вказується номер тесту, основної частини і еталонного значення відповіді. Основна частина тесту складається з питання / затвердження і трьох варіантів відповіді. При розробці варіантів відповідей необхідно реалізувати одне з наступних правил визначення еталонного значення відповіді:
  - це номер єдиного істинного варіанти відповіді;
  - це номер єдиного помилкового варіанту відповіді;
  - це 1 - якщо більшість варіантів відповідей справжні, і 0 - якщо більшість варіантів відповідей помилкові;

номер правила вказано в табл. 1;

- прийом даних від клієнта;
- визначення прізвища та відповідей тестованого користувача;
- вибір випадковим чином одного з тестів і передача його на сторону клієнта;
- число тестів, переданих користувачеві повинно бути не менше трьох;
- формування словника, що містить такі дані про процес тестування:
  - прізвище користувача, число виконаних тестів, число правильних відповідей;
  - прізвище та відповіді користувача, число правильно виконаних тестів;
  - прізвище та відповіді користувача, правильні відповіді;

номер вмісту словника для варіанту вказано в табл. 1;

- після закінчення тестування передача клієнту зібраних даних.

Табл.1

Номер п/п	Ввод имени	Ввод ответов	Вывод тестов	Тема тестов	Правило тестов	Данные словаря
12	prompt()	<input>		5)	3)	1)

Код програм

Файл Testing.py

```
#!C:\Users\Ham\AppData\Local\Programs\Python\Python36\python.exe

# -*- coding: utf-8 -*-

print ( "Access-Control-Allow-Origin: *" )

print ()
```

```
import cgi
import cgitb
import random
import pickle
cgitb.enable()
import sys
import codecs
from my_modd import write_results, D
cgitb.enable()
sys.stdout = codecs.getwriter('utf-8')(sys.stdout.detach())
sys.stdin = codecs.getreader('utf-8')(sys.stdin.detach())
def get_test (file_name): # Функция считывает из файла file_name тесты
        f= open (file_name, 'rb' ) # и возвращает один из них
        test_list=pickle.load(f)
        f.close()
        test=random.choice(test_list)
        test_list.remove(test)
        f= open ( 'cur_tests.dat' , 'wb' )
        pickle.dump(test_list,f)
        f.close()
        return test
max_test=3
data=cgi.parse() # Данные, полученные сервером:
if 'name' in data: # имя тестируемого
        name=data[ 'name' ][0]
        if name[: len (name)-1]== 'all' : # Вывод всех данных тестирования
                                f= open ( 'all_users.dat' , 'rb' )
                                while True :
                                                try :
                                                        user_dict=pickle.load(f)
                                                        if name[ len (name)-1]== '0' :
                                                                write_results(user_dict)
                                                        else :

        write_results(user_dict,tests=1,answers=1, points=1)
                                                        f.close()
                                                except EOFError : break
        else :
                        test=get_test( 'tests.dat' )
                        print (D,test.split( ';' )[0])
                        user_dict={ # Создание словаря данных пользователя:
                        'name' : data[ 'name' ][0], # имя
                        'number_test' : 1, # порядковй номер очередного теста
                        'tests' : [test], # список переданных тестов
                        'answers' : [], # список ответов пользователя
```

```
                    'points' : 0 } # число набранных баллов
                    f = open ( 'user.dat' , 'wb' )
                    pickle.dump(user_dict,f)
                    f.close()
if'answer' in data: # ответ тестируемого
                answer=data[ 'answer' ][0]
                f= open ( 'user.dat' , 'rb' )
                user_dict=pickle.load(f)
                user_dict[ 'answers' ]+=[answer]
                tests=user_dict[ 'tests' ]
                etalon=tests[ len (tests)-1].split( ';' )[1]
                if answer==etalon: user_dict[ 'points' ]+=1
                if user_dict[ 'number_test' ]<max_test:
                                test=get_test( 'cur_tests.dat' )
                                print (D,test.split( ';' )[0])
                                user_dict[ 'number_test'] +=1
                                user_dict[ 'tests'] +=[test]
                                f= open ( 'user.dat' , 'wb' )
                                pickle.dump(user_dict,f)
                                f.close()
                else :
                                write_results(user_dict,tests=1,answers=1,points=1)
                                f= open ( 'all_users.dat' , 'ab' )
                                pickle.dump(user_dict,f)
                                f.close()
# def write_results (user_dict, points=0, tests=0, answers=0):
        # print ('D,<i><b>RESULT TESTS:</b></i>' ,
                        # '<br>name &ndash; ' , user_dict[ 'name' ])
        # print("lol")
        # if tests:
                        # print ( '<br>TESTS &ndash; ' )
                        # for el in user_dict[ 'tests' ]: print (el.split( '.' )[0])
        # if answers:
                        # print ( '<br>answers &ndash; ' )
                        # for el in user_dict[ 'answers' ]: print (el)
        # if points:
                        # print ( '<br>ballov &ndash; ' ,
                        # user_dict[ 'points' ], ' iz ' , user_dict[ 'number_test' ])
        # print ( '<br>ocenka &ndash; ' )
        # m=user_dict[ 'aa' ]
        # if m==0: mark= "vv"
        # elif m==1: mark= "bb"
```

```
        # elif m==2: mark= "aaaa"
        # else : mark= "otlichno"
        # print ( "" , mark, "")
```

Файл Form_tests.py

```
#!C:\Users\Ham\AppData\Local\Programs\Python\Python36\python.exe
import pickle
TESTS=[ 'Тест №1.Какой результат будет после выполнения данной операции: 2 ** 3' \
        '<br>1.Число 2 возведётся в степень 3 <br> 2. 8 <br>3. 6;1' ,
        'Тест №2.Какой результат будет после выполнения данной операции: 2*Khpi' \
        '<br>1.Ошибка 3 <br> 2.KhpiKhpi <br>3. 8;0' ,
        'Тест №3.Какой результат будет после выполнения данной операции: 4 / 3' \
        '<br>1. 1 <br> 2.1(3)в периоде <br>3.1.3333333333333333;1' ,
        'Тест №4.Какой результат будет после выполнения данной операции: -4 // 3' \
        '<br>1. 1 <br>2. -2 <br>3. -1;0' ,
        'Тест №5.Какой результат будет после выполнения данной операции: 2 << 2' \
        #<!--'фукция func() будет непосредсвтвенно доступна программе?' \-->
        '<br>1. 8 <br>2. 2 <br>3. 1;0' \
        'Тест №6.Какой результат будет после выполнения данной операции: 5 & 3 даёт 1.?' \
'<br>1. 1 <br>2. 2 <br>3. 15 ;0' ]

f= open ( 'tests.dat' , 'wb' )
pickle.dump(TESTS,f)
f.close()
print ( 'Список тестов TESTS создан и записан в файл tests.dat.' )
```

.html

```
<html>
<head>
<title>Тестирование пользователя с использованием технологии AJAX</TITLE>
<link rel="stylesheet" href="../lab.css">
</head>
<body>
<h1>Тестирование</h1>
<button onclick="subm1()">Начать <br>тестирование</button>   
<button onclick="subm2()">Указать <br>ответ</button>
<input id="answer">
<div id='div' />
<script>
   var xhr=new XMLHttpRequest(),
      myDiv=myDiv=document.getElementById("div");
   function subm1() {
    myDiv.innerHTML='';
    var name=prompt('Введите фамилию, инициалы или all0/all1','');
    if(name) {
     xhr.open('GET','http://localhost/scripts/testing.py?r='+Math.random()+'&name='+name);
```

```
      xhr.send(null);
    }
  else alert("Имя не введено");
}
function subm2() {
  <!-- var answer=prompt('Введите ответ в виде трех двоичных цифр',");    -->
        var answer = document.getElementById('answer').value;
   if (answer) {
    xhr.open('GET','http://localhost/scripts/testing.py?r='+Math.random()+'&answer='+answer);
    xhr.send(null);
  }
  else alert("Ответ не указан")
}
    xhr.onerror=function(e) {
    myDiv.innerHTML+="<p style='color:red'>Ошибка XMLHttpRequest</p>";
 }
 xhr.onload=function(e) {
    myDiv.innerHTML=this.responseText;
 }
</script>
</body>
</html>
```

Результати програми

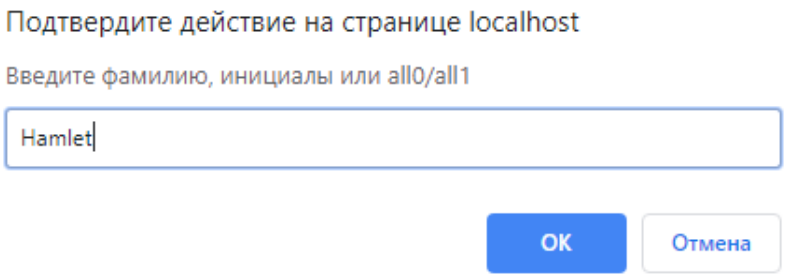

Рисунок 1 – введення імені

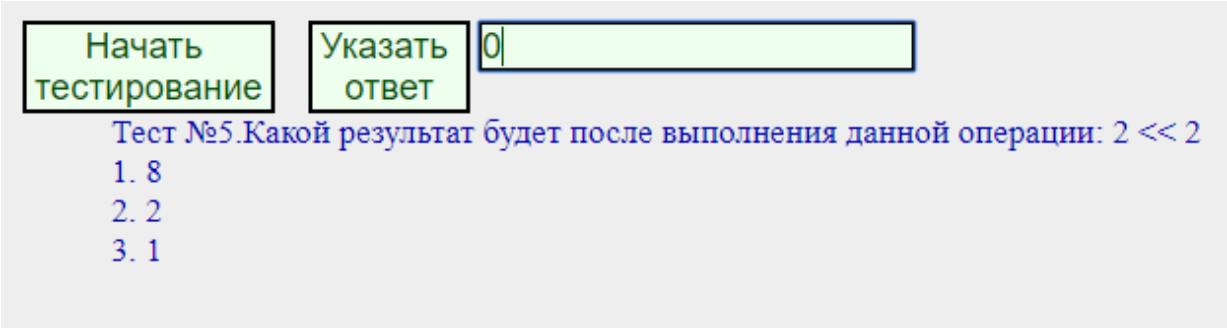

Рисунок 2 – тест

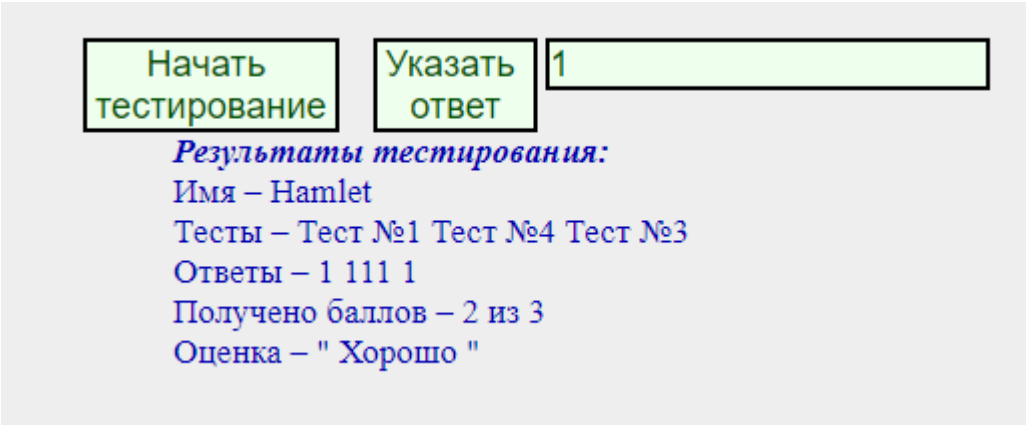

Рисунок 3 - результат тестування

Висновок

Розробили програму на базі технологій "клієнт-сервер" і AJAX, яка здійснює тестування користувача на стороні клієнта. Клієнтську частиау реалізували в вигляді HTML-документа з включеними фрагментами JavaScript, завданнями якого є: прийом імені користувача, передача на сервер імені користувача з використанням AJAX-технології, прийом з сервера тестів з використанням AJAX-технології та візуалізація їх на веб-сторінці,прийом відповідей користувача, прийом з сервера результатів тестування з використанням AJAX-технології та візуалізація їх на веб-сторінці (так само, як і тести). Розглянули особливості роботи серверного додатка, що використовує технологію AJAX

Лабораторна робота №11

Тема: Створення призначених для користувача класів

Мета: Розгляд способів створення класів, об'єктів і їх використання

ІНДИВІДУАЛЬНІ ЗАВДАННЯ

Розробити програму на мові Python, в якій:
створюється клас, що описує поведінку об'єктів, що становлять (див. колонку табл. 3) таких персонажів:
1 - користувачів комп'ютера;
2 - літературних персонажів;
3 - студентів;
4 - героїв мультиплікації;
5 - історичних персонажів;
6 - персонажів художніх фільмів;
клас повинен мати следулющіе спеціальні методи: __init __ (), __str __ () і __del __ ();
клас повинен мати такі атрибути і / або методи (див. колонку "Атрибути / методи" табл. 3):
1 - статичний метод;
2 - атрибут класу;
3 - метод екземпляра класу;
4 - закритий атрибут
5 - закритий метод
здійснюється управління двома атрибутами класу, для першого встановлюється режим "тільки читання", для другого - згідно колонці "Управління" табл. 3):
1 виконується читання атрибута і запис в нього;
2 полягає у зчитуванні та видалення атрибута;
3 виконується читання, запис і видалення атрибута;
створюються об'єкти класу і перевіряється їхня робота.

Код програми

```
class My_class:
        '''Поведение объектов, представляющих персонажей художественных
фильмов'''
        total=0

        def __init__(self,name):
                My_class.total+=1
                print( 'Джон,к нам пришли гости.','Как ваше имя ?',sep='
')
                self.__name=name
        def all_people(self):
                print('Всего людей',My_class.total)
        def public_method (self):
                print ( 'Это открытый метод' )
                self.all_people()
        def __str__(self):
```

```
                return "Меня зовут"+self.__name +'Я Джон, приятно
познакомиться! '
        def getName(self):
                print('Getting name')
                return self.__name
        def setName(self, value):
                print('Setting name to ' + value)
                self.__name = value
        def delName(self):
                print('Deleting name')
                del self.__name
        def __del__(self):
                print( 'Гость по имени' +self.__name + ' уходит'  )
        name = property(getName, setName, delName, 'Name property')
v1=My_class( ' Шерлок!' )
print(v1)
v2=My_class(" Hamlet!")
print(v2)
print(v2.public_method())
del v1
del v2
p = My_class('Sherlock')
print(p.name)
p.name = 'Hamlet'
del p.name
```

Робота програми

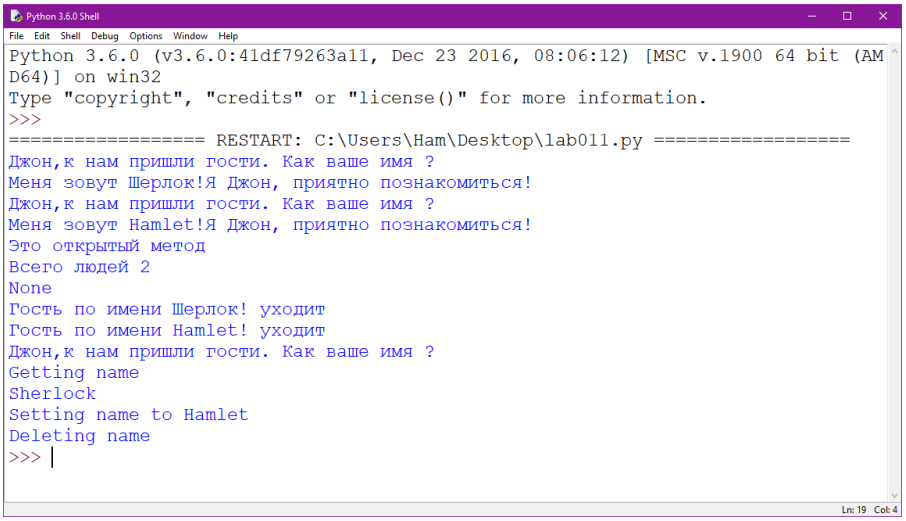

Рисунок 1 – робота програми

Висновок

Розглянули способі створення класів, об'єктів і їх використання

ЛАБОРАТОРНАЯ РАБОТА №12

Тема: Наследование классов

Цель: Розглядаються способи створення нових класів шляхом їх дослідження

Индивидуальные задания

Використовувати програму на мові Python, яку необхідно:

1. Створити класс, надіючий родительський (базовий) клас, в якості якого використовувати класс, розроблений в лаб. раб. №11 (цей класс описывает поведение персонажей). При цьому наследуемый класс повинен мати:

розширювальний (по відношенню до батьківського класу) конструктор (метод __init __ ());

перегружаемый метод;

новий метод (або методы);

2. створювати другий класс;

3. створити третій клас за допомогою множинного нагляду первого і другого класів (цей класс повинен містити методи як першого, так і другого класів);

4. Створити класс, який наследует неизвестный класс плавати і з допомогою спеціального методу __new __ () перетворює значення тривалості і веси, заданих в одній системі вимірювання, в іншому випадку колонкою "Преобразовать" табл. 1:

1 - з дюймов в міліметрі;

2 - из сатниметров в дюймы;

3 - з футов в метры;

4 - из метров в футы;

5 - из фунтов в килограммы;

6 - з граммов в караты;

7 - з килограммов в унції;

5. Створити класс у вигляді послідовності, що має бути методами, що вказуються в колонці "Методи" табл. 1:

1 - __init __ ();

2 - __getitem __ ();

3 - __setitem __ ();

4 - перший ();

5 - останній ();

6 - __iter __ ();

7 - __str __ ();

8 - __len __ ();

9 - push ();

10 - pop ().

Номер п/п	Преобразование	Методы
12	5	1,2,3,5,7,8

Код програми

```
from math import radians, sin, cos, acos

class My_class:
      '''Поведение объектов, представляющих персонажей художественных фильмов'''
      total=0
      def __init__(self,name,mood):
         My_class.total+=1
         print( '-Джон,к нам пришли гости. ','-Как ваше имя ?',sep='\n')
         self.name=name
         self.__mood=mood
      def talk(self):
         print('-Привет!Меня зовут '+self.name+ '\n-Какой род вашей
деятельности?\n'+self.__mood)
      def delName(self):
            print('Deleting name')
            del self.name
      def __del__(self):
            print( 'Гость по имени ' +self.name + ' уходит'  )
class My_pickle:
   ''' Сохранение экземпляров объектов на диске '''
   import pickle
   def __init__ (self,file):
      self.file=file
   def put (self,obj):
      f= open (self.file, 'wb' )
      My_pickle.pickle.dump(obj,f)
      f.close()
   def get (self):
            with open (self.file, 'rb' ) as f:
               return My_pickle.pickle.load(f)
class My_classTwo(My_class):
   "Поведение персонажов"
   def __init__(self,name,mood,where):
      print('Действия происходят в ',where,end="\n")
      super().__init__(name,mood)
   def talk(self):
      print("-I'm " +self.name)
   def tell(self,obj):
      print("-Кто-кто??")
      obj.talk()
      print(obj.name,"",sep="",end="")
      if self.name=="Мориарти":
         print(",привет, мой годами проверенный враг!")
```

```
        else:
            print(", Если вы не мой враг,то вы мой друг!")
s1=My_classTwo('Майкрофт Холмс','О господи, я всего лишь простой государственный служащий!','Белгравии. ')
s1.__class__=My_class
s1.talk()
s2=My_classTwo('Мориарти',' Преподаватель математики, Злодей-консультант,','Лондоне. ',)
s2.talk()
s2.tell(s2)
class Geocoordsys:
   "Класс - погода"
   def __init__(self,x,y):
      self.x=x
      self.y=y
      print('Широта:'+self.x,'Долгота:',self.y)
   def __call__(self,x,y):
      print('Широта:'+self.x,'Долгота:',self.y)
#we=Geocoordsys("55.755831","37.617673")
#Множественное наследование
class New_hero(My_classTwo,My_pickle):
        def __init__(self,name,mood,where,file):
                super().__init__(name,mood,where)
                self.file=file
        def tell(self,obj):
                print("А ты кто?")
                obj.talk()
                print(obj.name+",я пожалуй запишу в блокнот.")
                self.put(obj)
print("_____
_____\n")

m=My_classTwo('Ирэн',' Это секрет!','гостиной 221b.')
m.talk()
m.tell(m)
new1=New_hero('Hadson',' Домовладелица, секретарша и танцовщица','той же квартире.','obj2.dat')
new1.talk()
new1.tell(m)
s3=new1.get()
s3.__class__=My_class
#s3.talk() #IREN
#s3.name
class PoundstoKilograms(float):
        "Из фунтов в килограмы"
        def __new__(cls,arg):
```

```
            arg=arg/2.2046
            return float. __new__(cls,arg)
print('1 фут = ',PoundstoKilograms(1),'кг')

class My_sequence:
      "Последовательность.."
      def __init__ (self,values= None ) :
            if values is None :
                  self.values=[]
            else :
                  self.values=values
      def __getitem__ (self,key):
            ''' Получить значение по ключу '''
            return self.values[key]
      def __setitem__ (self,key,value):
            ''' Установить значение по ключу '''
            self.values[key]=value
      def last (self):
            ''' Получить значение последнего элемента '''
            return self.values[-1]
      def __str__ (self):
            ''' Возвратить строковое представление объекта '''
            return 'Герои: ' + str (self.values)
      def __len__ (self):
            ''' Возвратить длину последовательности '''
            return len (self.values)
h2=My_sequence(["Шерлок Холмс","Доктор Ватсон","Миссис Хадсон","Майкрофт", "Ирeэн Адлер","Профессор Мориарти"])
print('Последний герой:',h2.last())
print('Количество героев:',h2.__len__())
print(h2.__str__())
h2[3]='Грег' #setItem
print("GetItem:",h2[3])
```

Работа програми

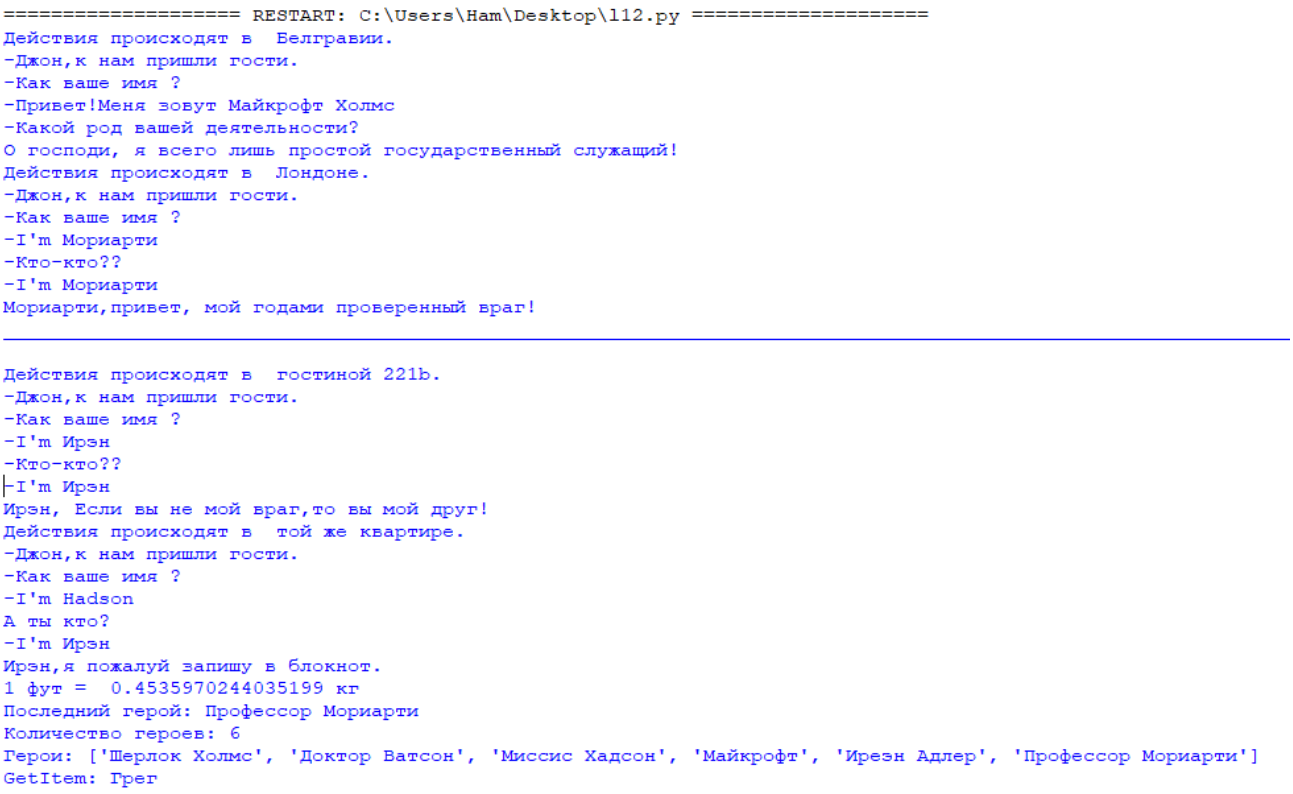

Рисунок 1 – робота програми

Висновок

Розглянули способи створення нових класів шляхом їх дослідження

Лабораторна робота №13

Тема: Розробка графічного інтерфейсу

Мета: Розгляд способів створення класів, об'єктів і їх використання

Индивидуальные задания

Розробити програму на мові Python, в якій на базі модуля tkinter розроблений графічний інтерфейс, який має кілька вікон:

вікно з кнопкою (об'єкт Button), напис на якій змінюється по кожному кліку і містить наступну інформацію - номер клацання і число, яке визначається згідно колонці "Число" табл. №1:

- 1 - чергове просте число (починаючи з 1);
- 2 - чергове число Фібоначчі (починаючи з 1);
- 3 - чергове просте число виду 2 n - 1 (починаючи з n = 1);

вікно, що містить:

текстове поле (об'єкт Text), куди виводиться початок деякого невеликого повідомлення (розповіді і т.п.);

поле введення пароля (об'єкт Entry);

кнопку для продовження виведення повідомлення (об'єкт Button);

друга текстове поле (об'єкт Text), куди виводиться несподіване (смішне) завершення повідомлення (при правильному введенні пароля) або повідомлення про неправильному введенні паролі;

мітки (об'єкт Label) - при необхідності;

вікно, що містить:

п'ять прапорців (об'єкти Checkbutton), що задають згідно колонці "Прапорці" табл. №1:

1 - числа 1, 2, 4, 8 і 16;

2 - 5 букв кирилиці (на розсуд студента);

2 - 5 англійських букв (на розсуд студента);

текстове поле, куди поміщається результат:

для чисел - сума заданих чисел;

для букв - слово, що складається з обраних букв;

вікно, що містить:

перемикачі (об'єкти Radiobutton), що задають згідно колонці "Перемикачі" табл. №1:

1 - змінні типи;

2 - незмінні типи;

3 - контейнери;

4 - послідовності;

текстове поле, куди поміщається результат - приклад литерала для обраного класу.

Таблиця 1 - Перелік індивідуальних завдань

Номер п/п	Число	Флажки	Переключатели
12	3	2	4

Код програми 13_1.py

```
from tkinter import *
def lucas_lehmer_test(p, m):
    s = 4
    for _ in range(p - 2):
        s = ((s * s) - 2) % m
    return s == 0
def prime_test_list(n, known_primes):
    if n < 2:
        return False
    if n % 2 == 0:
        return n == 2  # 2 is the only even prime
    k = 0
    while known_primes[k] ** 2 <= n:
        if n % known_primes[k] == 0:
            return False
        k += 1
    return True
def mersenne_prime_print_list(a):
    known_primes = [2]
    known_mersenne_primes = [3]

    for n in range(3, a, 2):
        if prime_test_list(n, known_primes):
            known_primes.append(n)
            m = 2 ** n - 1
            if lucas_lehmer_test(n, m):
                known_mersenne_primes.append(m)
    return known_mersenne_primes
class My_frame2(Frame):
        ''' Описание рамки с кнопкой, подсчитывающей события '''
        def __init__(self,master):
                super().__init__(master)
                self.grid()
                self.btn_clicks=0
                self.create_widgets()
        def create_widgets(self):
                ''' Создает кнопку с обработкой событий '''
                self.btn=Button(text='Число щелчков: 0',
                    command=self.count,
                    width='100',
                    height='12',
                    bg='#efffef',
                    font='"calibri light" 16',
                    fg='blue' )
                self.btn.grid()
        #def simple_number(self):
         #   self.btn_clicks = mersenne_prime_print_list(20)
         #   self.btn['text']='Число  '+str(self.btn_clicks)
```

```
    def count(self):

        self.btn_clicks+=1
        self.btn['text']='Число щелчков: '+str(self.btn_clicks)+\
        '\nMersenne: \n'+str(mersenne_prime_print_list(self.btn_clicks))
root=Tk()
root.title('Обработка событий')
root.geometry('150x550+800+200')
app=My_frame2(root)
root.mainloop()
```

Код програми 13_2.py

```
from tkinter import *
class My_frame(Frame):
    '''  Рамка для работы с текстом  '''
    def __init__(self,master):
        super(My_frame, self).__init__(master)
        self.grid()
        self.create_widgets()
    def create_widgets(self):
        ''' Создает 2 метки, текстовое поле, текстовую область и кнопку'''
    #Надпись-инструкция
        self.lb_ins=Label(self,text=Рассказ?',
            fg='blue', font='arial 14')
        self.lb_ins.grid(row=0, column=0, columnspan=2, sticky=W)
    #Надпись-пароль
        self.lb2=Label(self, text='Пароль',fg='blue', font='arial 10')
        self.lb2.grid(row=1, column=0, sticky=E)
    #текстовое поле для ввода пароля
        self.ent=Entry(self,bg='#f0f0ff', font='arial10')
        self.ent.grid(row=1, column=1, sticky=E)
    #создание текстовой области, куда будет помещен ответ
        self.txt2=Text(self, width=30, height=4, wrap=WORD, bg='#f0f0ff',
                font='arial 10')
        self.txt2.grid(row=3, column=0, columnspan=3)

        self.txt=Text(self, width=30, height=4, wrap=WORD, bg='#f0f0ff',
                font='arial 10')
        self.txt.grid(row=2, column=0, columnspan=4)

    #кнопка отправки сообщения
        self.bt=Button(self,
            text='Узнать секрет',
            fg='blue', bd='2', font='arial 12',
            command=self.reveal)
        self.bt.grid(row=0, column=2, sticky=E)
    def reveal(self):
        contents=self.ent.get()#Вы ошиблись номером
        if contents=='secret':
```

```
            message='Ответил знакомый голос'

        else:
            message='Вы ввели неправильный пароль.'
        self.txt2.delete(0.0,END)
        self.txt2.insert(0.0,message)


root=Tk()
root.title('Работа с текстом')
root.geometry('250x160+850+300')
root.resizable(True, False)
app=My_frame(root)
root.mainloop()
```

Код програми 13_3.py

```
from tkinter import *
class My_frame4(Frame):
    '''  Использование флажков  '''
    def __init__(self,master):
        super(My_frame4, self).__init__(master)
        self.grid()
        self.create_widgets()
    def create_widgets(self):
        ''' Создает шесть флажка '''
        #метка-описание
        Label(self,
          text='Моё имя', font='arial 13', fg='blue'
                ).grid(row=0, column=0,sticky=W)
        #метка-инструкция
        Label(self,
            text='Выберете все, что нравится', font='arial 11',
            fg='blue').grid(row=1, column=0, sticky=W)
        #флажок 'буква - 1'
        self.likes_letter1=BooleanVar()
        Checkbutton(self,text='Г',
            variable=self.likes_letter1,
            command=self.update_text
            ).grid(row=2, column=0,sticky=W)
        #флажок 'буква - 2'
        self.likes_letter2=BooleanVar()
        Checkbutton(self,
            text='а',
            variable=self.likes_letter2,
            command=self.update_text
            ).grid(row=3, column=0, sticky=W)
        #флажок 'буква - 3'
        self.likes_letter3=BooleanVar()
        Checkbutton(self,
```

```
                text='м',
                variable=self.likes_letter3,
                command=self.update_text
                ).grid(row=4, column=0, sticky=W)
            #флажок 'Буква - 4 '
            self.likes_letter4=BooleanVar()
            Checkbutton(self,
                text='л',
                variable=self.likes_letter4,
                command=self.update_text
                ).grid(row=5, column=0, sticky=W)
            #флажок 'Буква - 5 '

            self.likes_letter5=BooleanVar()
            Checkbutton(self,
                text='е',
                variable=self.likes_letter5,
                command=self.update_text
                ).grid(row=6, column=0, sticky=W)
            #флажок 'Буква - 6 '

            self.likes_letter6=BooleanVar()
            Checkbutton(self,
                text='т',
                variable=self.likes_letter6,
                command=self.update_text
                ).grid(row=7, column=0, sticky=W)
            #текстовая область с результатом
            self.results_txt=Text(self, fg='black', font='14', width=50, height=5, wrap=WORD)
            self.results_txt.grid(row=8, column=0, columnspan=3)
        def update_text(self):
            likes=""
            if self.likes_letter1.get():
                likes+='Г'
            if self.likes_letter2.get():
                likes+='а'
            if self.likes_letter3.get():
                likes+='м'
            if self.likes_letter4.get():
                likes+='л'
            if self.likes_letter5.get():
                likes+='е'
            if self.likes_letter6.get():
                likes+='т'
            self.results_txt.delete(0.0,END)
            self.results_txt.insert(0.0,likes)

root=Tk()
root.title('Использование флажков')
```

```
root.geometry('270x250+800+200')
app4=My_frame4(root)
root.mainloop()
```

Код програми 13_4.py

```
from tkinter import *
class My_frame5(Frame):
    '''  Использование флажков  '''
    def __init__(self,master):
        super(My_frame5, self).__init__(master)
        self.grid()
        self.create_widgets()
    def create_widgets(self):
        ''' Создает переключатель '''
        #метка-инструкция
        Label(self,
            text='Последовательности', font='arial 13', fg='blue'
            ).grid(row=0, column=0,sticky=W)
        self.favorite=StringVar()
        self.favorite.set(None)
        #положение переключателя 'String'
        Radiobutton(self,
            text='String',
            variable=self.favorite,
            value='Hi',
            command=self.update_text
            ).grid(row=1, column=0,sticky=W)
        #положение переключателя 'Lists'
        Radiobutton(self,
            text='Lists',
            variable=self.favorite,
            value=['one','two','three'],
            command=self.update_text
            ).grid(row=2, column=0,sticky=W)
        #положение переключателя 'Lists'
        Radiobutton(self,
            text='Tuples',
            variable=self.favorite,
            value=(1, 2, 3),
            command=self.update_text
            ).grid(row=3, column=0,sticky=W)
        Radiobutton(self,
            text='Dictionaries',
            variable=self.favorite,
            value= {"guava": "a tropical fruit", "python": "a programming language", "the
answer": 42},
            command=self.update_text
            ).grid(row=4, column=0,sticky=W)
```

```
        #текстовая область для результата
        self.results_txt=Text(self, fg='blue', font='arial 11', width=60,
                              height=4, wrap=WORD)
        self.results_txt.grid(row=5, column=0, columnspan=4)
    def update_text(self):
        message="Ваша последовательность - "
        message+=self.favorite.get()
        self.results_txt.delete(0.0,END)
        self.results_txt.insert(0.0,message)

root=Tk()
root.title('Использование переключателя')
root.geometry('310x220+800+200')
app4=My_frame5(root)
root.mainloop()
```

Робота програми

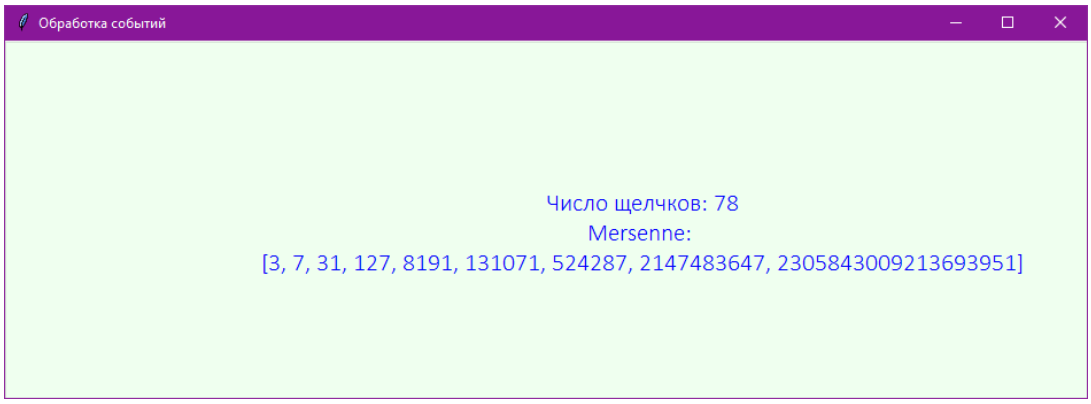

Рисунок 1 – програма 13_1.py

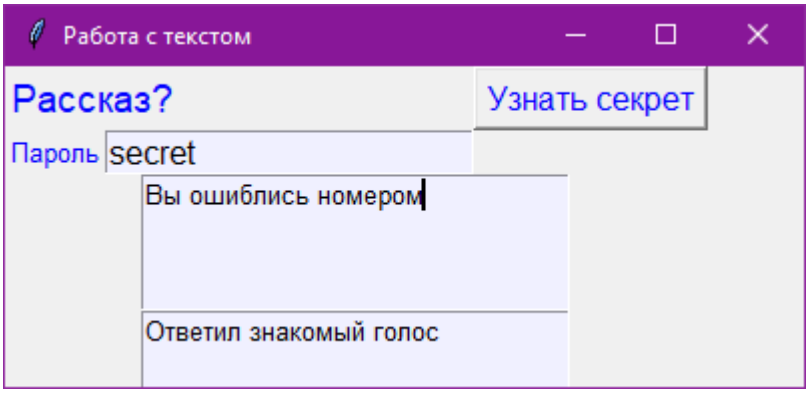

Рисунок 2 – програма 13_2.py

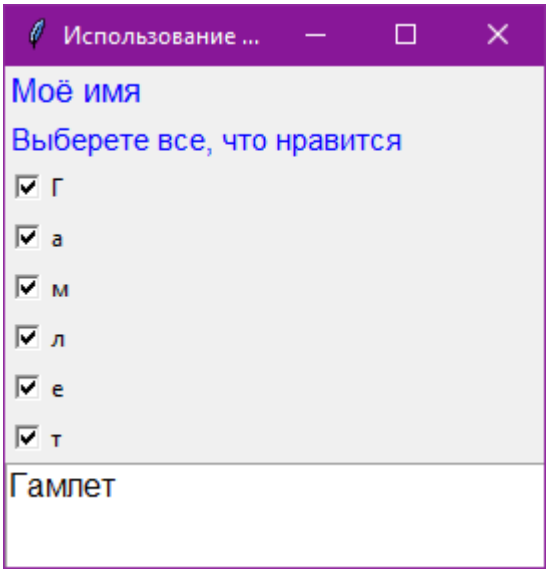

Рисунок 3 – програма 13_3.py

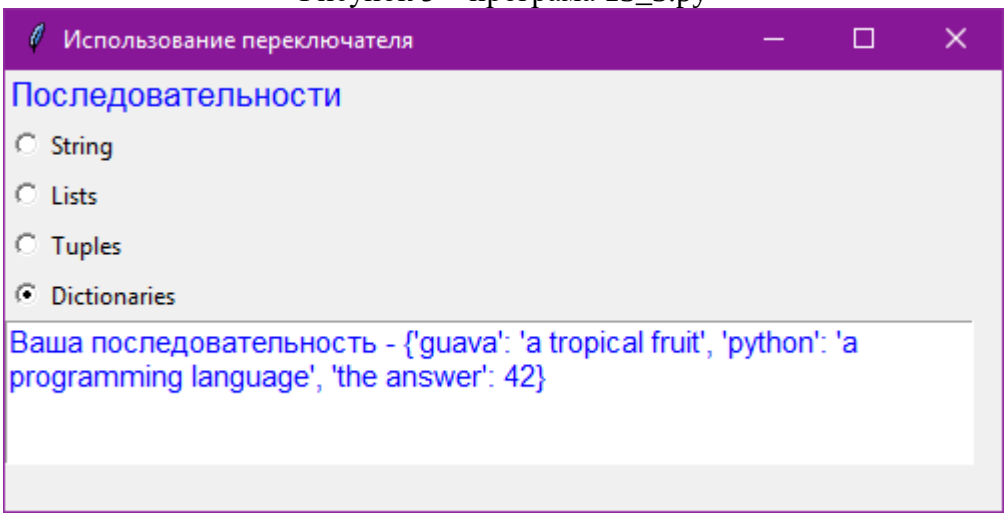

Рисунок 4 – програма 13_4.py

Висновок

Розробили програму на мові Python, в якій на базі модуля tkinter розроблений графічний інтерфейс, розглянули способі створення класів, об'єктів і їх використання.