

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования

«Вятский государственный университет»

(ФГБОУ ВО «ВятГУ»)

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Разработка программных систем

Применение С-кода в языке программирования Python

Вариант 3

Выполнил студент группы ИВТ-31 _____/Кудяшев Я.Ю./

Проверил преподаватель _____/Чистяков Г.А./

Киров 2022

1. Задание

В ходе выполнения лабораторной работы необходимо разработать анализатор строк; взаимодействие с пользователем должно осуществляться с применением пользовательского интерфейса.

2. Листинг программы

Листинг программы приведен в приложении А.

3. Тесты сравнения

Тесты были проведены для метода `number_of_words(line)`, который подсчитывает количество слов во входной строке.

Входные данные	C++	Python	Ускорение
500000 символов	0.001995 сек	0.053894 сек	27,01
1000000 символов	0.003989 сек	0.107715 сек	27,00
1500000 символов	0.004986 сек	0.161648 сек	32,42
2000000 символов	0.006984 сек	0.215405 сек	30,84
2500000 символов	0.006981 сек	0.270861 сек	38,80
		Максимальное	38,80
		Среднее	31,23
		Минимальное	27,00

4. Вывод

В ходе выполнения лабораторной работы был изучен основной функционал библиотеки `pybind11` для интеграции С-кода в Python. Проведено временное сравнение работы алгоритмов для работы со строками, реализованных на C++ и Python. Также были получены навыки для сборки интегрированного проекта в Visual Studio.

Приложение А
(обязательное)
Листинг программы

Lab_4_Python.py

```
import string_methods_c
import string_methods_python
from datetime import datetime
import time
import random
import string

def generate_random_string(length):
    letters = string.ascii_lowercase
    rand_string = ''.join(random.choice(letters + " ") for i in range(length))
    return rand_string

line = generate_random_string(2500000)
number = 500000

for i in range(5):

    print("Number of words in random string of length " + str(number))

    start_time = datetime.now()
    print(string_methods_c.number_of_words(line[:number]))
    print(datetime.now() - start_time)

    start_time = datetime.now()
    print(string_methods_python.number_of_words(line[:number]))
    print(datetime.now() - start_time)

    number+=500000
```

string_methods_python.py

```
def number_of_words(line):
    return sum(map(lambda x : int(x.isalpha()), line.split()))

def number_of_defined_characters(line,symbol):
    return line.count(symbol)

def number_of_symbols(line):
    return len(line)

def number_of_universal_symbols(line):
    return len(set(line))
```

string_methods_c.cpp

```
#include <pybind11/pybind11.h>
#include <string>
#include <set>

namespace py = pybind11;
using namespace std;

int number_of_words(string line)
{
    int count = 0;
    int p = line.length(), i = 0;
    while (i < p)
    {
        if (line[i] != ' ' && (line[i + 1] == ' ' || line[i + 1] == '\\0'))
            count++;
        i++;
    }
    return count;
}

int number_of_symbols(string line)
{
    int count = 0;
    int len = line.length();

    for (int i = 0; i < len; i++)
    {
        if (line[i] != ' ')
            count++;
    }
    return count;
}

int number_of_universal_symbols(string line)
{
    set<char> x;
    for (int i = 0; i < line.length(); i++)
        x.insert(line[i]);
    return x.size();
}

int numbrer_of_defined_characters(string line, char symbols)
{
    int count = 0;

    for (unsigned int i = 0; i < line.length(); i++)
        if (line[i] == symbols)
            count++;
    return count;
}

PYBIND11_MODULE(string_methods_c, m) {
    m.def("number_of_words", &number_of_words);
    m.def("number_of_symbols", &number_of_symbols);
    m.def("number_of_universal_symbols", &number_of_universal_symbols);
    m.def("numbrer_of_defined_characters", &numbrer_of_defined_characters);
}
```