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 C-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.

1. Листинг программы

Листинг программы приведен в приложении А.

1. 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 |

1. Вывод

В ходе выполнения лабораторной работы был изучен основной функционал библиотеки pybind11 для интеграции C-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);

}