Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи №2 з дисципліни
« Основи програмування 2. Модульне програмування»

«Бінарні файли»
Варіант 3

Виконав студент ІП-13, Баран Софія Володимирівна
(шифр, прізвище, ім'я, по батькові)

Перевірила Вєчерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота №1
Текстові файли
Варіант 3
Задача

3. Створити файл з інформацією про наявність побутової техніки в офісі: найменування, дата покупки, термін гарантії (у днях). Визначити, чи є в офісі побутова техніка на гарантії. Перенести в інший файл інформацію про техніку, у якої закінчився термін гарантії.

Код

Python

main.py:

```
from functions import *

todays_date = read_todays_date()

text_to_write = capture_text()
write_empty_file("input.txt", text_to_write)

text = read_file("input.txt")
print(f"Text on input:\n{text}\n")

lines = split_lines(text)
print("Lines list:")
print(*lines, sep='\n')

information = split_information(lines)
print("\nInformation list:")
print(*lines, sep='\n')

check_warranty(information, todays_date)

output_text = read_file("output.txt")
print("\n")
print(output_text)
```

functions.py:

```
import sys
import datetime
import time

def read_todays_date():
    date = input("Enter today's date (in format dd.mm.yy): ")

    return date

def capture_text():
    print('Enter your text\nPress Enter to go to the next
line\nPress Ctrl + D - to end writing')

    text = sys.stdin.read()

    # відкидаємо зайвий символ \n
    text = text[:len(text) - 1]

    print(f"Whole text:\n{text}\n")

    return text


def read_file(file_name: str):
    read_input_file = open(file_name, 'rt')
    text = read_input_file.read()
    read_input_file.close()

    return text


def split_lines(text: str):
```

```
    lines = text.split('\n')

    for i in range(len(lines)):
        if lines[i] == '':
            lines.pop(i)

    return lines


def split_information(lines: list[str]):
    general_information = []

    for i in range(len(lines)):
        information_array = lines[i].split(',')
        general_information.append(information_array)

    return general_information


def write_to_file(file_name: str, string_to_write: str):
    write_file = open(file_name, 'at')
    write_file.write(string_to_write)
    write_file.close()


def check_warranty(information: list[list[str]], todays_date: str):
    for i in range(len(information)):
        buy_date = information[i][1]
        buy_date_datetime = datetime.datetime.strptime(buy_date,
'%d.%m.%y')
        buy_date_unix = time.mktime(buy_date_datetime.timetuple())

        warranty_term = int(information[i][2])
        warranty_expire_date = buy_date_unix + warranty_term *
86400

        todays_datetime = datetime.datetime.strptime(todays_date,
'%d.%m.%y')
        todays_date_unix = time.mktime(todays_datetime.timetuple())

        if todays_date_unix > warranty_expire_date:
```

```
            print(f"Warranty have been expired for product
{information[i][0]}")

            string_to_write = f"Warranty have been expired for
product:
{information[i][0]},{information[i][1]},{information[i][2]}\n"
            write_to_file("output.txt", string_to_write)


def write_empty_file(file_name: str, text: str):
    write_file = open(file_name, 'wt')
    write_file.write(text)
    write_file.close()
```

C++

main.cpp:

```
#include "functions.h"

int main() {
    string todays_date = read_today_date();
    string text_to_write = capture_text();
    write_empty_file("input.txt", text_to_write);
    string text = read_input_file("input.txt");
    printf("\nText on input:\n%s\n\n", text.c_str());
    vector<string> lines = split_lines(text);
    print_vector(lines);

    vector<vector<string>> information = split_information(lines);

    check_warranty(information, todays_date);
    read_output_file("output.txt");
}
```

functions.h:

```
#include <iostream>
#include <sstream>
#include <iomanip>
#include <fstream>
#include <vector>
#include <ctime>
using namespace std;

string capture_text();
void write_empty_file(const string& file_name, const string& text);
string read_input_file(const string& file_name);
void print_vector(const vector<string>& vec);
vector<string> split_lines(const string& text);
vector<vector<string>> split_information(const vector<string>& lines);
void write_file(const string& file_name, const string& string_to_write);
int get_unix_timestamp(const string& day,const string& month,const string& year);
void check_warranty(vector<vector<string>> information, const string& todays_date);
void read_output_file(const string& file_name);
string read_today_date();
```

functions.cpp:

```
#include "functions.h"

string read_today_date(){
    string todays_date;
    cout << "Enter today's date (in format dd.mm.yy): ";
    getline(cin, todays_date);

    return todays_date;
}

string capture_text(){
    cout << "Enter your product info by the following format: Name,DD.MM.YY,Days\nPress
Enter to go to the next line\nPress '<' - to end writing\n";

    string text;
    string line = "";
    while(line != "<"){
        getline(cin, line);
        text += line + '\n';
    }
    text = text.substr(0, text.size()-2);

    return text;
}

void write_empty_file(const string& file_name, const string& text){
    ofstream write_file(file_name, ios::binary);
    write_file << text;
    write_file.close();
}

string read_input_file(const string& file_name){
    ifstream read_file(file_name, ios::binary);
    string text;

    while (!read_file.eof()){
        text += read_file.get();
    }
    read_file.close();

    text = text.substr(0, text.size()-1);

    return text;
}

void print_vector(const vector<string>& vec){
    cout << "Lines vector:\n";

    for(auto & i : vec){
        cout << i << "\n";
    }
    cout << "\n";
}

vector<string> split_lines(const string& text){
    vector<string> lines;
    int previous_position = 0;
    int counter = 0;

    while(counter <= text.size()){
        if(text[counter] == '\n' || counter == text.size()){
            string line = text.substr(previous_position, counter-previous_position);
            lines.push_back(line);
            previous_position = counter+1;
        }
        counter++;
```

```
    }

    return lines;
}

vector<vector<string>> split_information(const vector<string>& lines){
    vector<vector<string>> general_information;

    for(int i=0;i<lines.size();i++){
        int previous_position = 0;
        int counter = 0;
        vector<string> information_vector;

        while(counter <= lines[i].size()){
            if(lines[i][counter] == ',' || counter == lines[i].size()){
                string info = lines[i].substr(previous_position, counter-
previous_position);
                information_vector.push_back(info);
                previous_position = counter+1;
            }
            counter++;
        }

        general_information.push_back(information_vector);
    }

    return general_information;
}

void write_file(const string& file_name, const string& string_to_write){
    ofstream write_file(file_name, ios::app | ios::binary);
    write_file << string_to_write;

    write_file.close();
}

int get_unix_timestamp(const string& day,const string& month,const string& year){
    tm time{};

    stringstream ss("20" + year + "-" + month + "-" + day + "T00:00:00.000Z");

    ss >> get_time(&time, "%Y-%m-%dT%H:%M:%S");
    time_t time_stamp = mktime(&time);

    return static_cast<int>(time_stamp);
}

void check_warranty(vector<vector<string>> information, const string& todays_date){
    for(int i=0; i < information.size(); i++){
        string buy_date = information[i][1];
        string buy_date_day = buy_date.substr(0, 2);
        string buy_date_month = buy_date.substr(3, 2);
        string buy_date_year = buy_date.substr(6, 2);
        int buy_date_unix = get_unix_timestamp(buy_date_day, buy_date_month,
buy_date_year);

        int warranty_term = stoi(information[i][2]);
        int warranty_expire_date = buy_date_unix + warranty_term * 86400;

        string todays_date_day = todays_date.substr(0, 2);
        string todays_date_month = todays_date.substr(3, 2);
        string todays_date_year = todays_date.substr(6, 2);
        int todays_date_unix = get_unix_timestamp(todays_date_day, todays_date_month,
todays_date_year);

        if(todays_date_unix > warranty_expire_date){
            printf("Warranty have been expired for product: %s\n",
information[i][0].c_str());
            string string_to_write = "Warranty have been expired for product: " +
information[i][0] + "," + information[i][1] + "," + information[i][2] + "\n";
```

```
            write_file("output.txt", string_to_write);
        }

    }
}

void read_output_file(const string& file_name){
    ifstream read_output_file(file_name, ios::binary);
    cout << "\nOutput file:\n" << read_output_file.rdbuf() << "\n";
    read_output_file.close();
}
```

Результат програми:

```
Enter today's date (in format dd.mm.yy):28.03.22
 Enter your product info by the following format: Name,DD.MM.YY,Days
Press Enter to go to the next line
Press Ctrl + D - to end writing
tv,10.12.21,10
fridge,15.10.21,20
test,24.02.22,5
toaster,25.03.22,10
something,10.02.22,10
keyboard,27.03.22,5
^D

Text on input:
tv,10.12.21,10
fridge,15.10.21,20
test,24.02.22,5
toaster,25.03.22,10
something,10.02.22,10
keyboard,27.03.22,5

Lines vector:
tv,10.12.21,10
fridge,15.10.21,20
test,24.02.22,5
toaster,25.03.22,10
something,10.02.22,10
keyboard,27.03.22,5
Warranty have been expired for product tv
Warranty have been expired for product fridge
Warranty have been expired for product test
Warranty have been expired for product something

Output file:
Warranty have been expired for product: tv,10.12.21,10
Warranty have been expired for product: fridge,15.10.21,20
Warranty have been expired for product: test,24.02.22,5
Warranty have been expired for product: something,10.02.22,10


Process finished with exit code 0
```

```
Enter today's date (in format dd.mm.yy): 28.03.22
Enter your text
Press Enter to go to the next line
Press Ctrl + D - to end writing
tv,10.12.21,10
fridge,15.10.21,20
test,24.02.22,5
smartphone,25.03.22,10
table,10.02.22,10
laptop,27.03.22,5
^D
Whole text:
tv,10.12.21,10
fridge,15.10.21,20
test,24.02.22,5
smartphone,25.03.22,10
table,10.02.22,10
laptop,27.03.22,5

Text on input:
tv,10.12.21,10
fridge,15.10.21,20
test,24.02.22,5
smartphone,25.03.22,10
table,10.02.22,10
laptop,27.03.22,5
```

```
Lines list:
tv,10.12.21,10
fridge,15.10.21,20
test,24.02.22,5
smartphone,25.03.22,10
table,10.02.22,10
laptop,27.03.22,5

Information list:
tv,10.12.21,10
fridge,15.10.21,20
test,24.02.22,5
smartphone,25.03.22,10
table,10.02.22,10
laptop,27.03.22,5
Warranty have been expired for product tv
Warranty have been expired for product fridge
Warranty have been expired for product test
Warranty have been expired for product table


Warranty have been expired for product: tv,10.12.21,10
Warranty have been expired for product: fridge,15.10.21,20
Warranty have been expired for product: test,24.02.22,5
Warranty have been expired for product: table,10.02.22,10
```