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.»

“Класи та об’єкти”

Варіант: 30

Виконав студент ІП-11 Тихонов Федір Сергійович

(шифр, прізвище, ім'я, по батькові)

Перевірив

(прізвище, ім'я, по батькові)

**Лабораторна робота №2**

**Класи та об’єкти**

**Мета -** вивчити механізми створення і використання класів і об’єктів.

**Постановка задачі -** Розробити клас, який є абстракцією тексту і підтримує операції додавання рядка до тексту і пошуку найдовшого рядка у тексті. Створити масив об’єктів даного класу. Доповнити декілька текстів рядками. Визначити найкоротший із найдовших рядків.

**Реалізація на С++:**

Header

#pragma once

#include <iostream>

#include <string>

#include <vector>

class TextClass{

public:

TextClass();

std::string findLongest();

void appendLine(std::string line);

void outputText();

std::vector<std::string> getText();

void setText(std::vector<std::string> text);

~TextClass();

private:

std::vector<std::string> text\_;

};

std::vector<TextClass> createTexts();

void outputTexts(std::vector<TextClass> texts);

void askToFindLongestString(std::vector<TextClass> texts);

void askToAppendLines(std::vector<TextClass>& texts);

void outputShortestOfLongest(std::vector<TextClass>(texts));

Source

#include "lib.h"

TextClass::TextClass() {

text\_ = {};

}

void TextClass::appendLine(std::string line) {

text\_.push\_back(line);

}

std::string TextClass::findLongest() {

std::string longestLine;

for(const std::string& line : text\_){

if(longestLine.length() < line.length()){

longestLine = line;

}

}

return longestLine;

}

void TextClass::outputText() {

std::cout << std::endl << "Outputting the contents of the text:" << std::endl;

for(const std::string& line : text\_){

std::cout << line << std::endl;

}

std::cout << std::endl;

}

TextClass::~TextClass() {

text\_.clear();

}

std::vector<std::string> TextClass::getText() {

std::vector<std::string> text;

for(const std::string& line : text\_){

text.push\_back(line);

}

return text;

}

void TextClass::setText(std::vector<std::string> text) {

text\_.clear();

for(const std::string& line : text){

text\_.push\_back(line);

}

}

std::vector<TextClass> createTexts(){

std::vector<TextClass> texts;

std::string stopPushingTexts = "n";

std::string stopPushingLines = "n";

std::string line;

int textCounter = 1;

while(stopPushingTexts != "y"){

TextClass text;

std::cout << "Pushing lines for text " << textCounter << std::endl;

std::string stopPushingLines = "n";

while(stopPushingLines != "y"){

std::cout << "Enter a line" << std::endl;

if(textCounter > 1){

std::cin.ignore();

}

std::getline(std::cin, line);

text.appendLine(line);

std::cout << "If you want to stop pushing lines, type 'y'." << std::endl;

std::getline(std::cin, stopPushingLines);

}

std::cout << "Stop pushing texts? y/n?" << std::endl;

std::cin >> stopPushingTexts;

texts.push\_back(text);

textCounter++;

}

return texts;

}

void outputTexts(std::vector<TextClass> texts){

int textCounter = 0;

for(TextClass text : texts){

textCounter++;

std::cout << "Text" << textCounter << std::endl;

text.outputText();

}

}

void askToFindLongestString(std::vector<TextClass> texts){

std::string answer;

std::cout << "Do you want to know the longest string in a certain text?\nType 'yes' if you do.\n";

std::cin >> answer;

if(answer == "yes"){

int index;

std::cout << "Enter a number of text: ";

std::cin >> index;

index--;

std::cout<< texts[index].findLongest() << std::endl;

}

}

void askToAppendLines(std::vector<TextClass>& texts){

std::string permissionForTexts;

int i;

int counter = 0;

std::string line;

std::string permissionForLines;

std::cout << "Do you want to append lines to a text?" << std::endl << "Type 'yes' if you do." << std::endl;

std::cin.ignore();

std::getline(std::cin, permissionForTexts);

while(permissionForTexts == "yes"){

std::cout << "Enter a number of text: ";

std::cin >> i;

i--;

permissionForLines = "X";

while(permissionForLines != "STOP"){

std::cout << "Enter a line" << std::endl;

std::cin.ignore();

std::getline(std::cin, line);

texts[i].appendLine(line);

std::cout << "Type 'STOP' to stop pushing lines." << std::endl;

std::getline(std::cin, permissionForLines);

counter++;

}

texts[i].outputText();

std::cout << std::endl << "The longest string in the updated text is: " << std::endl;

std::cout << texts[i].findLongest() << std::endl;

std::cout << "Do you want to keep pushing lines for texts? Type \"yes\" to keep pushing lines for texts." << std::endl;

std::getline(std::cin, permissionForTexts);

}

}

void outputShortestOfLongest(std::vector<TextClass>(texts)){

std::string buffer = texts[0].findLongest();

std::string comparable;

int textNum = 1;

int counter = 1;

for(TextClass text : texts){

comparable = text.findLongest();

if(buffer.length() > comparable.length()){

buffer = comparable;

textNum = counter;

}

counter++;

}

std::cout << "The shortest line in text out of longest strings is in text № " << textNum << std::endl;

std::cout << buffer << std::endl;

}

Main

#include "lib.h"

int main() {

std::vector<TextClass> texts = createTexts();

outputTexts(texts);

askToFindLongestString(texts);

askToAppendLines(texts);

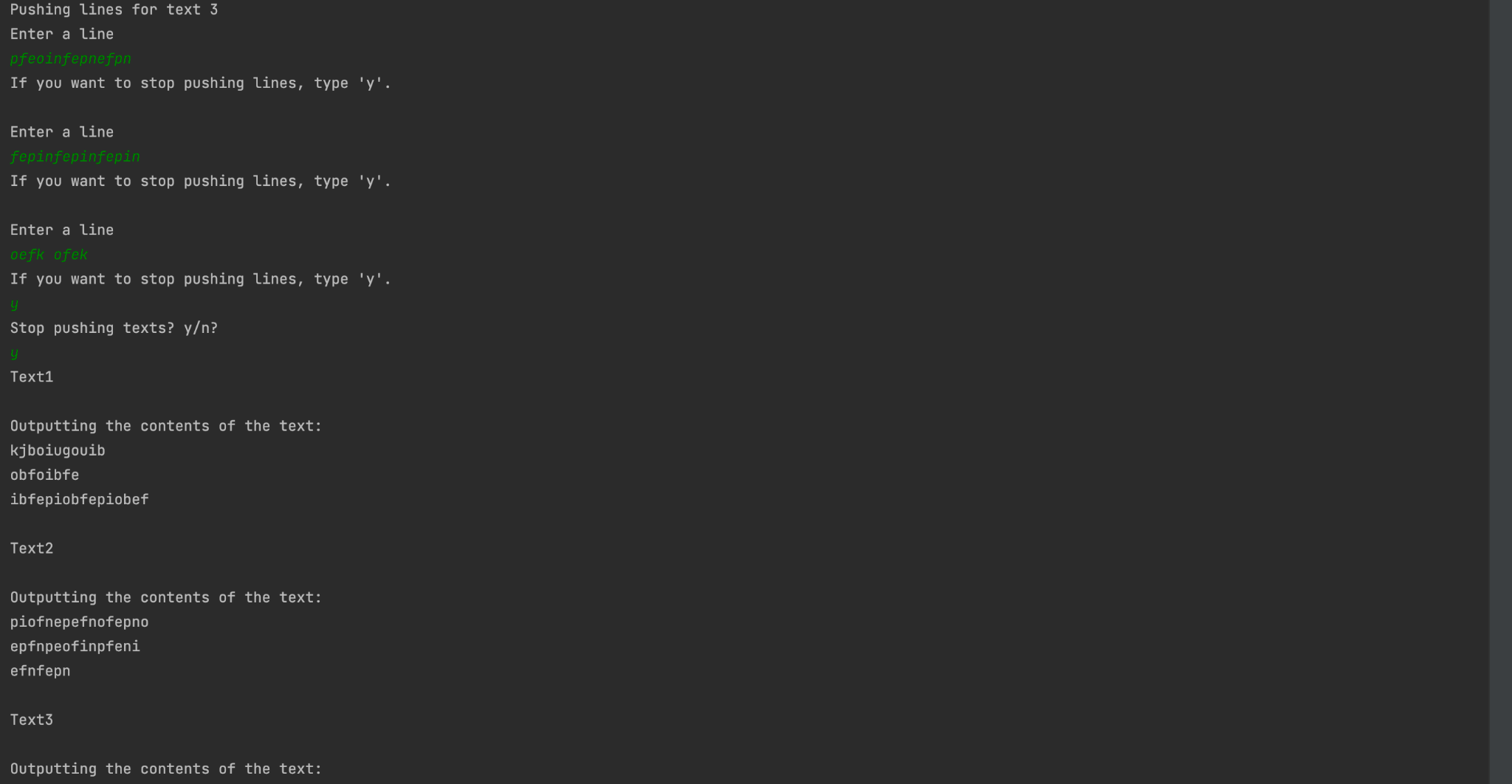
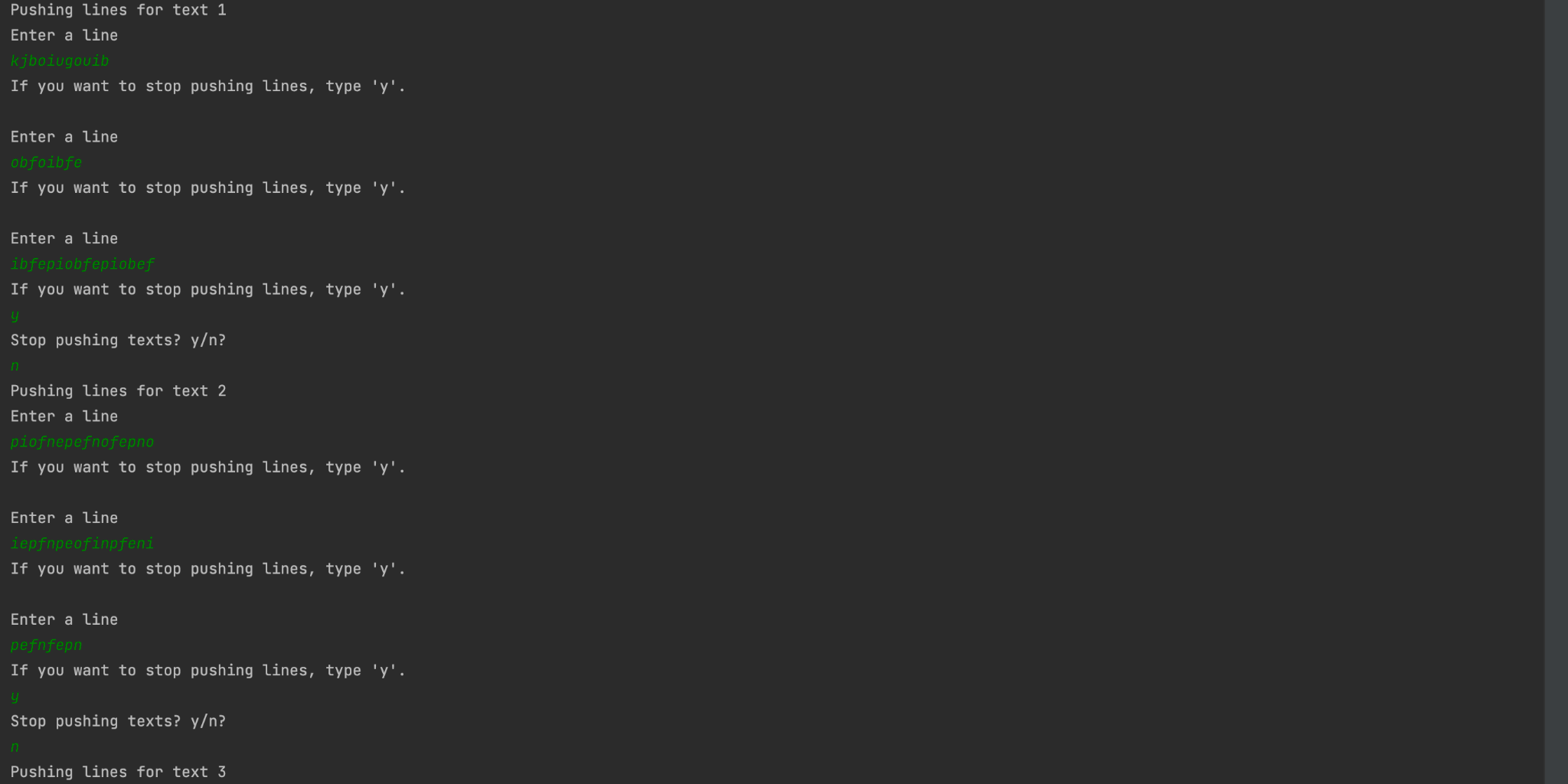
outputShortestOfLongest(texts);

outputTexts(texts);

return 0;

}

**Результати коду:**

****

**Висновок:**

Отже ми навчилися організовувати класи, ініціалізувати об’єкти та надавати властивості класам(атрибути і методи).