ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ

за курсом «Об’єктно-орієнтоване програмування»

студента групи ПЗ-21-2

Гудимова Дениса Сергійовича

кафедра математичного забезпечення ЕОМ, ДНУ

2022/2023 навч.р.

1. Постановка задачі

1. Налагодження тривіальної програми “Hello, World!” на мові C++.

Продемонструвати можливості налагодження, а саме: запуск програми у

покроковому режимі, різновиди точок зупинки, умовні точки зупинки, перегляд

значень змінних.

2. Настроювання інтегрованого середовища. Робота з проєктами.

Збереження проєкту у бажану папку. Перенос проєкту з комп’ютера на комп’ютер.

Зміна параметрів середовища (кольори, клавіатурні комбінації, тощо). Зміна

параметрів проєкту (ключі компілятора, параметри командного рядка, тощо).

3. Багатомодульні програми.

Скласти програму, яка викликає функції, визначені в інших (які окремо

транслюються) файлах. Вміти відкривати/закривати проєкт, вміти додавати новий

файл до існуючого проєкту.

4. Проєктування ієрархії та структури класів.

Спираючись на будь-яку предметну галузь за власним вибором, створити

ієрархію класів, що відображатиме сутності, представлені в цій галузі. Описати

спроєктовані класи словесно (згадуючи при цьому всі методи та поля

спроєктованих класів, їхню область видимості та призначення) або за допомогою

діаграми класів.

Продемонструвати 3 принципи ООП: інкапсуляція, наслідування та

поліморфізм (перевантаження та перевизначення).

2. Опис розв’язку

**Налагодження тривіальної програми**

1. Постановка задачі

Налагодження тривіальної програми “Hello, World!” на мові C++.

Продемонструвати можливості налагодження, а саме: запуск програми у

покроковому режимі, різновиди точок зупинки, умовні точки зупинки, перегляд

значень змінних.

2. Опис розв’язку

Існують точки зупнинкі звичайні та з умовою зупинки. Для низ є інструменти які надають можливості заходити у метод, проходити через метод та виходити з методу. Продивлятись значення змінних можна наводячи на них ао додаючи елемент до watch

3. Вихідний текст програми розв’язку задачі

#include "Function.h"

#include <iostream>

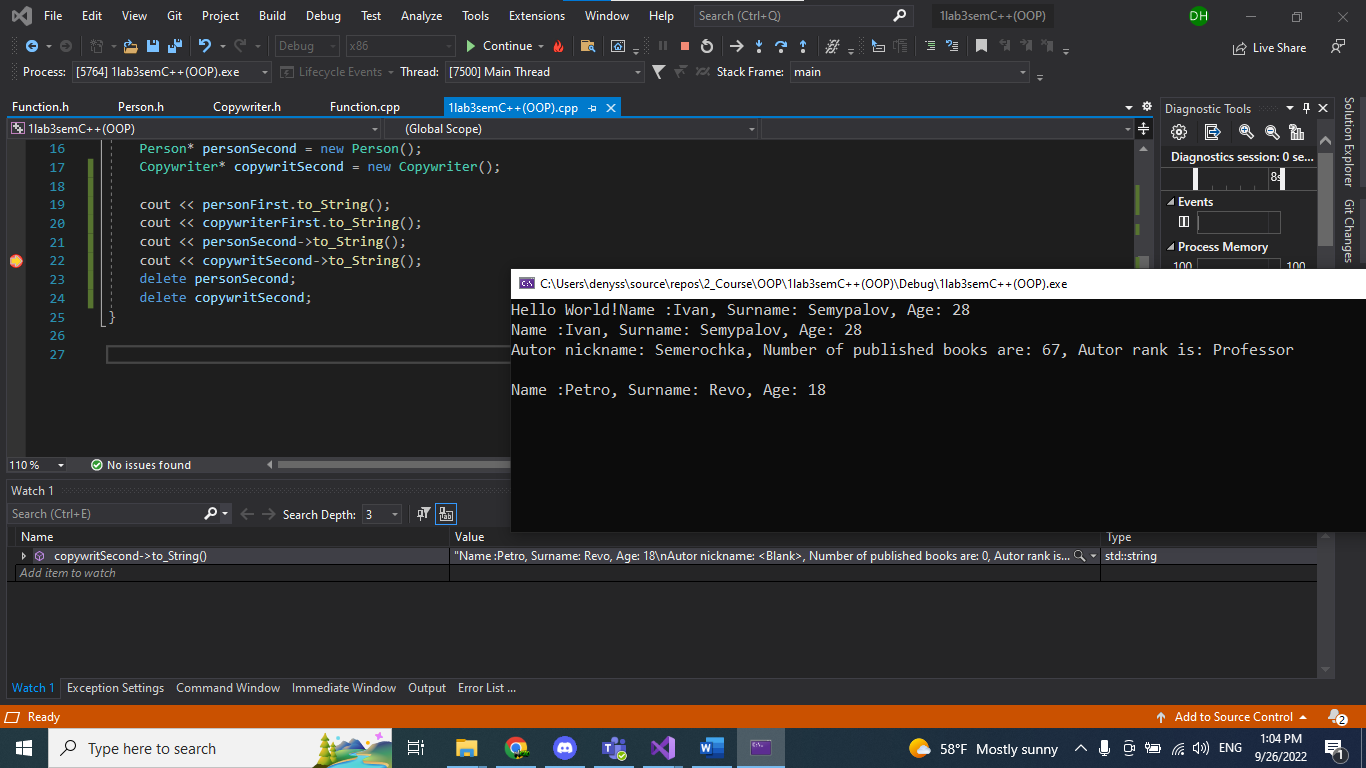
using namespace std;

cout << "Hello World!";

4. Опис інтерфейсу (керівництво користувача)

Існують точки зупнинкі звичайні та з умовою зупинки. Для низ є інструменти які надають можливості заходити у метод, проходити через метод та виходити з методу. Продивлятись значення змінних можна наводячи на них ао додаючи елемент до watch

5. Опис тестових прикладів

6. Аналіз помилок (опис усунення зауважень)

**Настроювання інтегрованого середовища**

1. Постановка задачі

Настроювання інтегрованого середовища. Робота з проєктами.

Збереження проєкту у бажану папку. Перенос проєкту з комп’ютера на комп’ютер.

Зміна параметрів середовища (кольори, клавіатурні комбінації, тощо). Зміна

параметрів проєкту (ключі компілятора, параметри командного рядка, тощо).

2. Опис розв’язку

Переходимо в Tools>Options>Environment>General для зміни теми IDE. Переходимо в Tools>Options>Environment>Keyboard для зміни клавіатурної комбінації в IDE.

Переходимо в Solution Explorer>right click on project>Properties>General для встановлення потрібних ключівкомпілятора.

Переходимо в Solution Explorer>right click on project>Properties>Debagging>Command Argument для встановлення параметрів командного рядка.

3. Вихідний текст програми розв’язку задачі

-

4. Опис інтерфейсу (керівництво користувача)

Переходимо в Tools>Options>Environment>General для зміни теми IDE. Переходимо в Tools>Options>Environment>Keyboard для зміни клавіатурної комбінації в IDE.

Переходимо в Solution Explorer>right click on project>Properties>General для встановлення потрібних ключівкомпілятора.

Переходимо в Solution Explorer>right click on project>Properties>Debagging>Command Argument для встановлення параметрів командного рядка.

5. Опис тестових прикладів

-

6. Аналіз помилок (опис усунення зауважень)

-

**Багатомодульні програми**

1. Постановка задачі

Багатомодульні програми.

Скласти програму, яка викликає функції, визначені в інших (які окремо

транслюються) файлах. Вміти відкривати/закривати проєкт, вміти додавати новий

файл до існуючого проєкту.

2. Опис розв’язку

В Function.h об’явлено прототип функції а у Function.cpp описана функція яка вікликаеться в main.

3. Вихідний текст програми розв’язку задачі

Function.h :

#pragma once

void printHW();

Function.cpp :

#include "Function.h"

#include <iostream>

using namespace std;

void printHW() {

cout << "Hello World!";

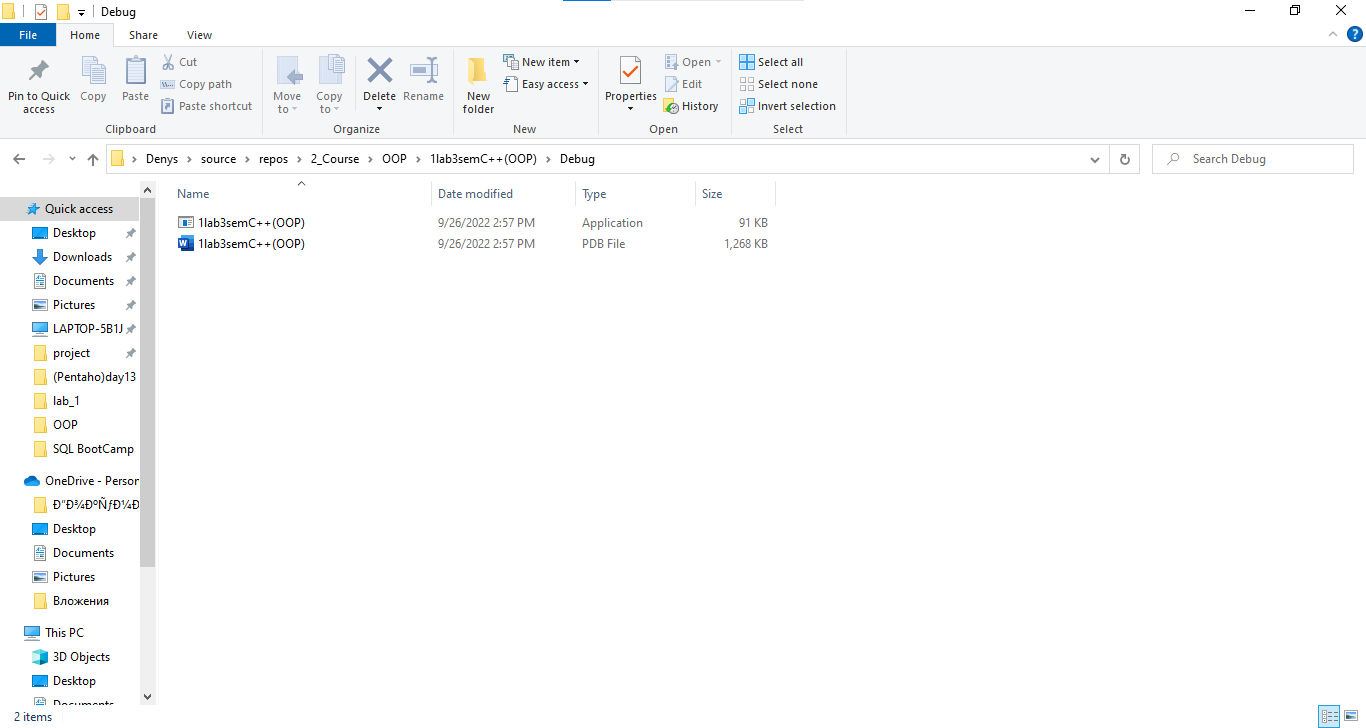
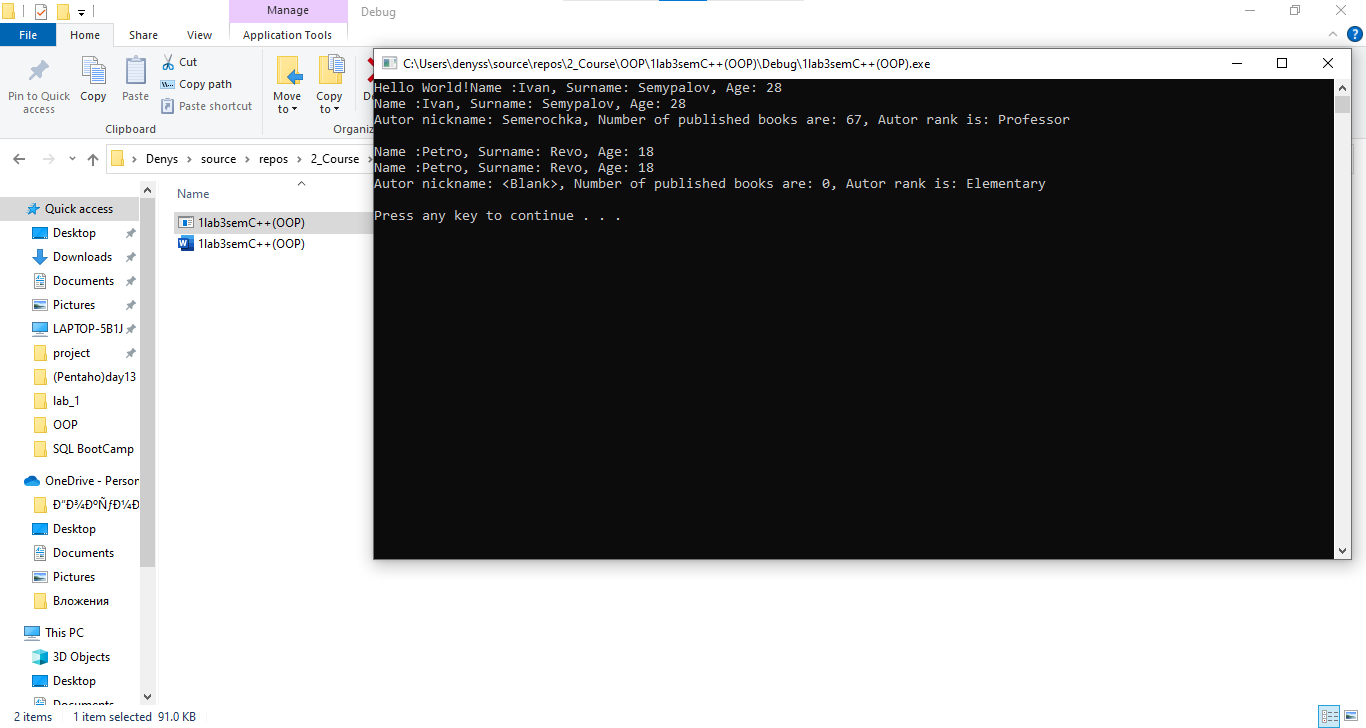
}

4. Опис інтерфейсу (керівництво користувача)

Відкрити папку з проектом зайти у папку Debug і запустити .exe файл.

Щоб закрити файл треба натиснути будьяку кнопку

5. Опис тестових прикладів

  6. Аналіз помилок (опис усунення зауважень)

**Проєктування ієрархії та структури класів**

1. Постановка задачі

Спираючись на будь-яку предметну галузь за власним вибором, створити

ієрархію класів, що відображатиме сутності, представлені в цій галузі. Описати

спроєктовані класи словесно (згадуючи при цьому всі методи та поля

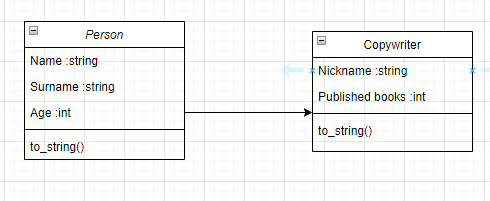
спроєктованих класів, їхню область видимості та призначення) або за допомогою

діаграми класів.

Продемонструвати 3 принципи ООП: інкапсуляція, наслідування та

поліморфізм (перевантаження та перевизначення).

2. Опис розв’язку



Copywriter наслідуе Person та переписуе метод to\_string() такж в обох классах реалізовані конструктори та property(get, set)

3. Вихідний текст програми розв’язку задачі

#pragma once

#include <string.h>;

#include <string>;

using namespace std;

class Person

{

private:

string name;

string surname;

int age;

public:

Person() {

setName("Petro");

setSurname("Revo");

setAge(18);

}

Person(string name, string surname, int age) {

setName(name);

setSurname(surname);

setAge(age);

}

//Geter

string getName() {

return name;

}

//Setter

void setName(string n) {

name = n;

}

//Geter

string getSurname() {

return surname;

}

//Setter

void setSurname(string surN) {

surname = surN;

}

//Geter

int getAge () {

return age;

}

//Setter

void setAge(int n) {

if (n > 0) {

age = n;

}

else {

age = 1;

}

}

virtual string to\_String() {

string s = "Name :" + getName() + ", Surname: " + getSurname() + ", Age: " + to\_string(getAge()) + "\n";

return s;

}

};

4. Опис інтерфейсу (керівництво користувача)

5. Опис тестових прикладів

6. Аналіз помилок (опис усунення зауважень)

Реалізовано 2 хедер файлы у яких описані батьківський клас Person та похідний Copywriter. Person містить 3 приватних поля 3 властивості и перевантажений метод ToString(). Copywriter Person містить 2 приватних поля 2 властивості и первизначений метод ToString().

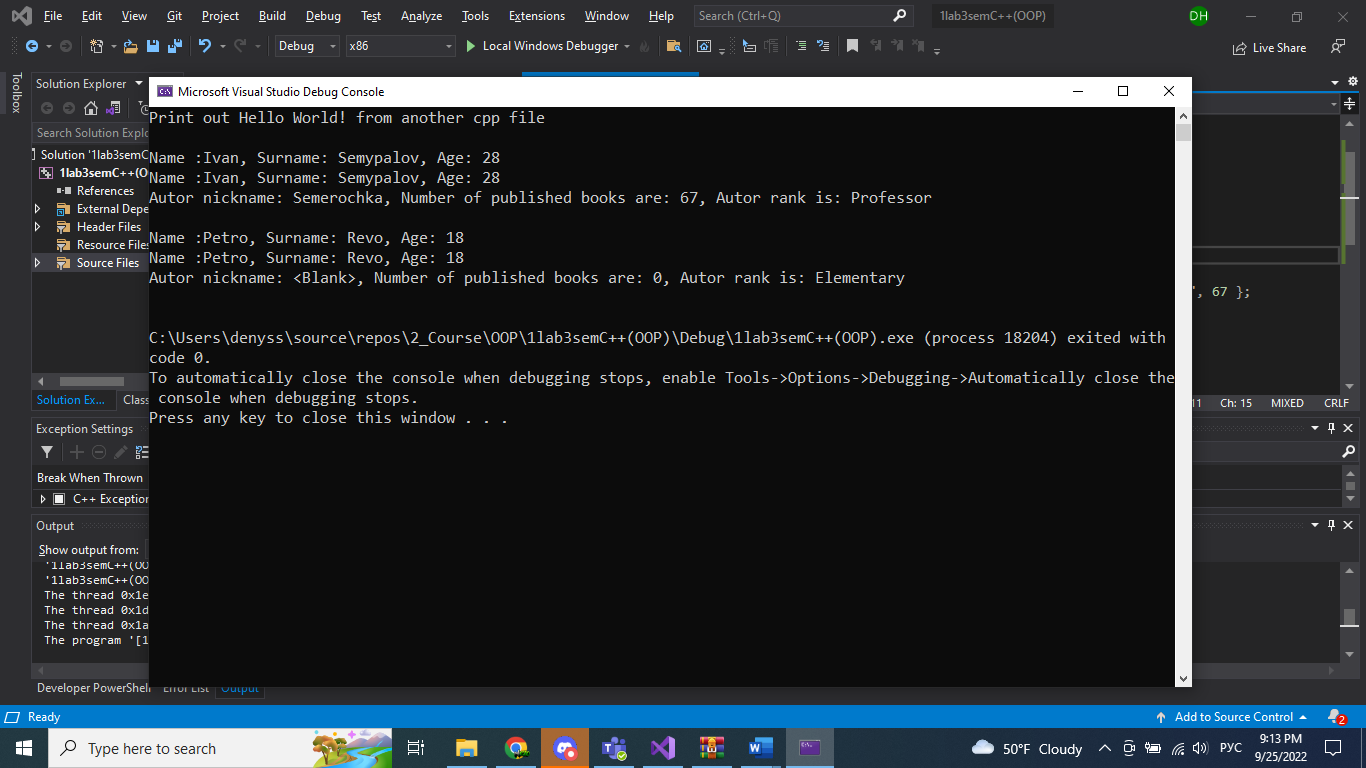
3. Вихідний текст програми розв’язку задачі

Додаеться 3 едер файли Person, Copywriter and Function

4. Опис інтерфейсу (керівництво користувача)

Інтерфейс відсутній

5. Опис тестових прикладів



6. Аналіз помилок (опис усунення зауважень)

What is this?