Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра програмного забезпечення



**ЗВІТ**

**Про виконання лабораторної роботи № 8**

«Наслідування. Створення та використання ієрархії класів»

**з дисципліни «***Об’єктно–орієнтоване програмування***»**

**Лектор:**

ст. викладач кафедри ПЗ

Коротєєва Т.О.

**Виконав:**

студ. групи ПЗ-12

Артех О.Ю.

**Прийняла:**

асистент кафедри ПЗ

Терендій О.В.

«\_\_\_» \_\_\_\_\_\_ 2020 р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів – 2020

**Тема роботи:** Наслідування. Створення та використання ієрархії класів.

**Мета роботи:** Навчитися створювати базові та похідні класи, використовувати наслідування різного типу доступу, опанувати принципи використання множинного наслідування. Навчитися перевизначати методи в похідному класі, освоїти принципи такого перевизначення.

**Теоретичні відомості**

*Наслідуванням* називається процес визначення класу на основі іншого класу. На новий (дочірній) клас за замовчуванням поширюються всі визначення змінних екземпляра і методів зі старого (батьківського) класу, але можуть бути також визначені нові компоненти або «перевизначені» визначення батьківських функцій і дано нові визначення. Прийнято вважати, що клас А успадковує свої визначення від класу В, якщо клас А визначений на основі класу В зазначеним способом.

*Класи можуть бути пов'язані один з одним різними відношеннями.* Одним з основних є відношення *клас-підклас*, відоме в об'єктно-орієнтованому програмуванні як *наслідування*. Наприклад, клас автомобілів Audi 6 є підкласом легкових автомобілів, який в свою чергу входить у більший клас автомобілів, а останній є підкласом класу транспортних засобів, який крім автомобілів включає в себе літаки, кораблі, потяги і т.д. Прикладом подібних відношень є системи класифікації в ботаніці та зоології.

При наслідуванні всі атрибути і методи батьківського класу успадковують-ся **класом-нащадком**. Наслідування може бути багаторівневим, і тоді класи, що знаходяться на нижніх рівнях ієрархії, успадкують всі властивості (атрибути і методи) всіх класів, прямими або непрямими нащадками яких вони є.

Крім одиничного, існує і **множинне** наслідування, коли клас наслідує відразу кілька класів (рис. 1). При цьому він успадкує властивості всіх класів, нащадком яких він є.

При наслідуванні одні методи класу можуть заміщатися іншими. Так, клас транспортних засобів буде мати узагальнений метод руху. У класах-нащадках цей метод буде конкретизований: автомобіль буде їздити, літак – літати, корабель – плавати. Така зміна семантики методу називається *поліморфізмом*. **Поліморфізм** – це виконання методом з одним і тим же ім'ям різних дій залежно від контексту, зокрема, від приналежності до того чи іншого класу. У різних мовах програмування поліморфізм реалізується різними способами.

**Параметр рівня доступу при наслідуванні**

При наслідуванні члени базового класу стають членами похідного класу. Як правило, для наслідування використовується наступна синтаксична конструкція.

сlass імя\_похідного\_класу: рівень\_доступу імя\_базового\_класу

{

// тіло класу

}

Рівень доступу визначає статус членів базового класу в похідному класі. Як цей параметр використовуються специфікатори public, private або protected. Якщо рівень доступу не вказаний, то для похідного класу за умовчанням використовується специфікатор private, а для похідної структури - public.

**Індивідуальне завдання**

1. Розробити ієрархію класів відповідно до варіанту.
2. Розробити базовий, похідні класи.
3. Використати public, protected наслідування.
4. Використати множинне наслідування (за необхідності).
5. Виконати перевантаження функції print() в базовому класі, яка друкує назву відповідного класу, перевизначити її в похідних. В проекті при натисканні кнопки виведіть на форму назви всіх розроблених класів.
6. Реалізувати методи варіанта та результати вивести на форму.

Створити базовий клас ДРУКОВАНА ПРОДУКЦІЯ (задаються назва, кількість сторінок).

Створити похідні класи КНИГА (задаються дані про автора, розмір сторінки, кількість

примірників) та ЖУРНАЛ (задаються розмір сторінки, кількість примірників,

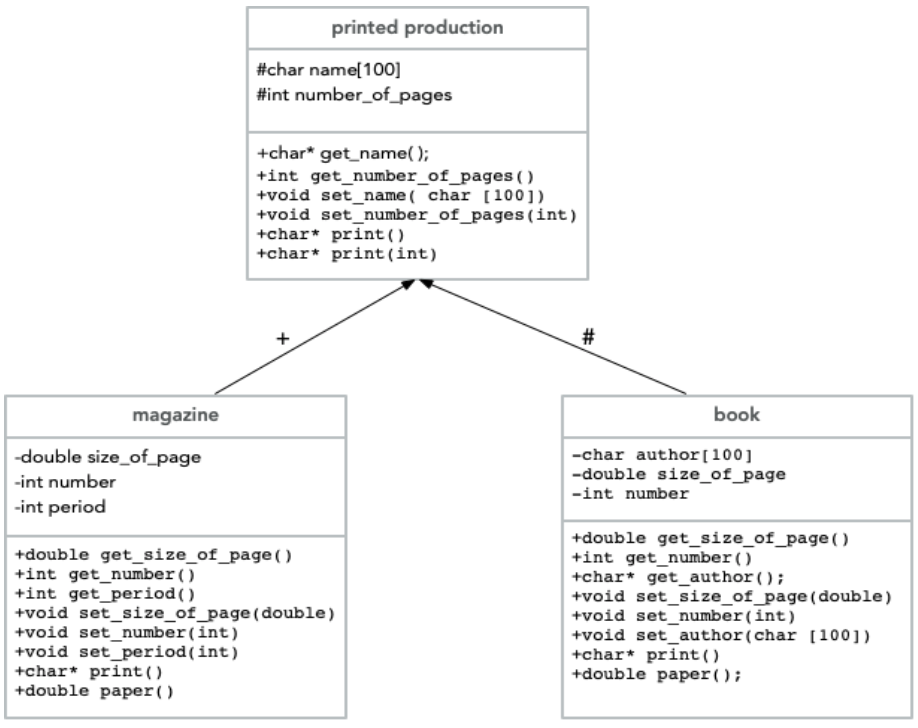
періодичність виходу). Для введених даних про друковану продукцію обчислити витрати

паперу (у кв. м.) – для кожного прикладу.

1. Оформити звіт до лабораторної роботи. Включити у звіт Uml-діаграму розробленої ієрархії класів.

**Хід роботи**

Розроблена діаграма:



Код програми:

**class.h**

#ifndef CLASS\_H

#define CLASS\_H

#include <iostream>

#include <string>

*class* **printed\_production**

{

*protected*:

char name[100]={'\0'};

int number\_of\_pages{};

*public*:

**printed\_production**()= *default*;; *//* *+*

**printed\_production**(*const* char name[100], int number\_of\_pages); *//* *+*

~**printed\_production**()= *default*;;

char\* **get\_name**(); *//* *+*

int **get\_number\_of\_pages**(); *//* *+*

void **set\_name**(*const* char name[100]); *//* *+*

void **set\_number\_of\_pages**(int number\_of\_pages); *//* *+*

char\* **print**(); *//* *+*

char\* **print**(int num); *//* *+*

};

*class* **magazine** : *public* printed\_production

{

*private*:

double size\_of\_page{};

int number{};

int period{};

*public*:

**magazine**()= *default*;;

~**magazine**()= *default*;;

**magazine**(*const* char name[100], int number\_of\_pages, double size\_of\_page, int number, int period);

double **get\_size\_of\_page**();

int **get\_number**();

int **get\_period**();

void **set\_size\_of\_page**(double size\_of\_page);

void **set\_number**(int number);

void **set\_period**(int period);

char\* **print**();

double **paper**();

};

*class* **book** : *protected* printed\_production

{

*private*:

char **auth**or[100]={'\0'};

double size\_of\_page{};

int number{};

*public*:

**book**()= *default*;;

~**book**()= *default*;;

**book**(*const* char name[100], int number\_of\_pages, double size\_of\_page, int number, *const* char author[100]);

cm

};

#endif *//* *CLASS\_H*

**mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

QT\_BEGIN\_NAMESPACE

*namespace* **Ui** { *class* **MainWindow**; }

QT\_END\_NAMESPACE

*class* **MainWindow** : *public* QMainWindow

{

Q\_OBJECT

*public*:

**MainWindow**(QWidget \*parent = *nullptr*);

~***MainWindow***();

*private* slots:

void **on\_print\_class\_names\_clicked**();

void **on\_magazine\_set\_clicked**();

void **on\_class2\_calc\_clicked**();

void **on\_book\_set\_clicked**();

void **on\_class3\_calc\_clicked**();

*private*:

Ui::MainWindow \*ui;

};

#endif *//* *MAINWINDOW\_H*

**class.cpp**

#include "class.h"

printed\_production::**printed\_production**(*const* char \*name, int number\_of\_pages) {

strcpy(*this*->name, name);

*this*->number\_of\_pages = number\_of\_pages;

}

char\* printed\_production::**print**() {

char \*str = *new* char[100];

str = "printed\_production";

*return* str;

}

char\* printed\_production::**print**(int num) {

char \*str = *new* char[100];

str = "printed\_production with num";

*return* str;

}

char\* printed\_production::**get\_name**(){

*return* name;

}

int printed\_production::**get\_number\_of\_pages**(){

*return* number\_of\_pages;

}

void printed\_production::**set\_name**(*const* char name[100]) {

strcpy(*this*->name, name);

}

void printed\_production::**set\_number\_of\_pages**(int number\_of\_pages) {

*this*->number\_of\_pages = number\_of\_pages;

}

*//-------------------------------------------------------------------------------------*

magazine::**magazine**(*const* char \*name, int number\_of\_pages, double size\_of\_page, int number, int period) {

strcpy(*this*->name, name);

*this*->number\_of\_pages = number\_of\_pages;

*this*->size\_of\_page = size\_of\_page;

*this*->number = number;

*this*->period = period;

}

double magazine::**get\_size\_of\_page**() {

*return* size\_of\_page;

}

int magazine::**get\_number**() {

*return* number;

}

int magazine::**get\_period**() {

*return* period;

}

void magazine::**set\_size\_of\_page**(double size\_of\_page) {

*this*->size\_of\_page = size\_of\_page;

}

void magazine::**set\_number**(int number) {

*this*->number = number;

}

void magazine::**set\_period**(int period) {

*this*->period = period;

}

char\* magazine::**print**() {

char \*str = *new* char[8];

str = "magazine";

*return* str;

}

double magazine::**paper**(){

*return* number\_of\_pages\*size\_of\_page\*number;

}

*//-------------------------------------------------------------------------------------*

book::**book**(*const* char \*name, int number\_of\_pages, double size\_of\_page, int number,*const* char author[100]) {

strcpy(*this*->name, name);

*this*->number\_of\_pages = number\_of\_pages;

*this*->size\_of\_page = size\_of\_page;

*this*->number = number;

strcpy(*this*->author,author);

}

double book::get\_size\_of\_page() {

*return* size\_of\_page;

}

int book::get\_number() {

*return* number;

}

char\* book::get\_author() {

*return* author;

}

void book::set\_size\_of\_page(double size\_of\_page) {

*this*->size\_of\_page = size\_of\_page;

}

void book::set\_number(int number) {

*this*->number = number;

}

void book::set\_author(*const* char author[100]) {

strcpy(*this*->author, author);

}

char\* book::print() {

char \*str = *new* char[4];

str = "book";

*return* str;

}

double book::paper(){

*return* number\_of\_pages\*size\_of\_page\*number;

}

**main.cpp**

#include "mainwindow.h"

#include <QApplication>

int **main**(int argc, char \*argv[])

{

QApplication a(*argc*, argv);

MainWindow w;

w.show();

*return* a.exec();

}

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include "class.h"

printed\_production temp;

magazine temp\_magazine;

book temp\_book;

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(parent)

, ui(*new* Ui::MainWindow)

{

ui->setupUi(*this*);

}

MainWindow::~***MainWindow***()

{

*delete* ui;

}

void MainWindow::**on\_print\_class\_names\_clicked**()

{

ui->class1\_print\_name->setText(temp.print());

ui->class2\_print\_name->setText(temp\_magazine.print());

ui->class3\_print\_name->setText(temp\_book.print());

}

void MainWindow::**on\_magazine\_set\_clicked**()

{

QString temp = ui->magazine\_name->text();

std::string temp1 = temp.toStdString();

char temp2[100];

std::strcpy(temp2,temp1.c\_str());

temp\_magazine.set\_name(temp2);

temp\_magazine.set\_number\_of\_pages(ui->magazine\_numbe\_of\_pages->text().toInt());

temp\_magazine.set\_size\_of\_page(ui->magazine\_size\_of\_page->text().toDouble());

temp\_magazine.set\_number(ui->magazine\_number->text().toInt());

temp\_magazine.set\_period(ui->magazine\_period->text().toInt());

ui->class2\_name\_2->setText(temp\_magazine.get\_name());

ui->class2\_number\_of\_pages->setNum(temp\_magazine.get\_number\_of\_pages());

ui->class2\_size\_of\_page->setNum(temp\_magazine.get\_size\_of\_page());

ui->class2\_number->setNum(temp\_magazine.get\_number());

ui->class2\_period->setNum(temp\_magazine.get\_period());

}

void MainWindow::**on\_class2\_calc\_clicked**()

{

ui->class2\_res->setNum(temp\_magazine.paper());

}

void MainWindow::**on\_book\_set\_clicked**()

{

QString temp = ui->book\_name->text();

std::string temp1 = temp.toStdString();

char temp2[100];

std::strcpy(temp2,temp1.c\_str());

temp\_book.set\_name(temp2);

temp\_book.set\_number\_of\_pages(ui->book\_number\_of\_pages->text().toInt());

temp\_book.set\_size\_of\_page(ui->book\_size\_of\_page->text().toDouble());

temp\_book.set\_number(ui->book\_number->text().toInt());

temp = ui->book\_author->text();

temp1 = temp.toStdString();

std::strcpy(temp2,temp1.c\_str());

temp\_book.set\_author(temp2);

ui->class3\_name->setText(temp\_book.get\_name());

ui->class3\_number\_of\_pages->setNum(temp\_book.get\_number\_of\_pages());

ui->class3\_size\_of\_page->setNum(temp\_book.get\_size\_of\_page());

ui->class3\_number->setNum(temp\_book.get\_number());

ui->class3\_author->setText(temp\_book.get\_author());

}

void MainWindow::**on\_class3\_calc\_clicked**()

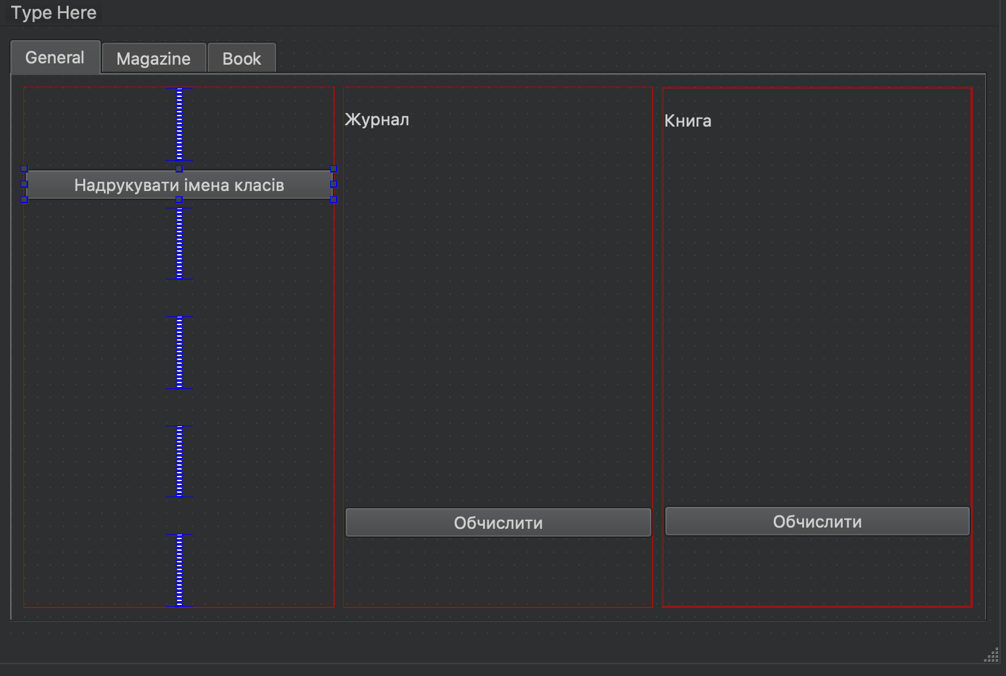
{

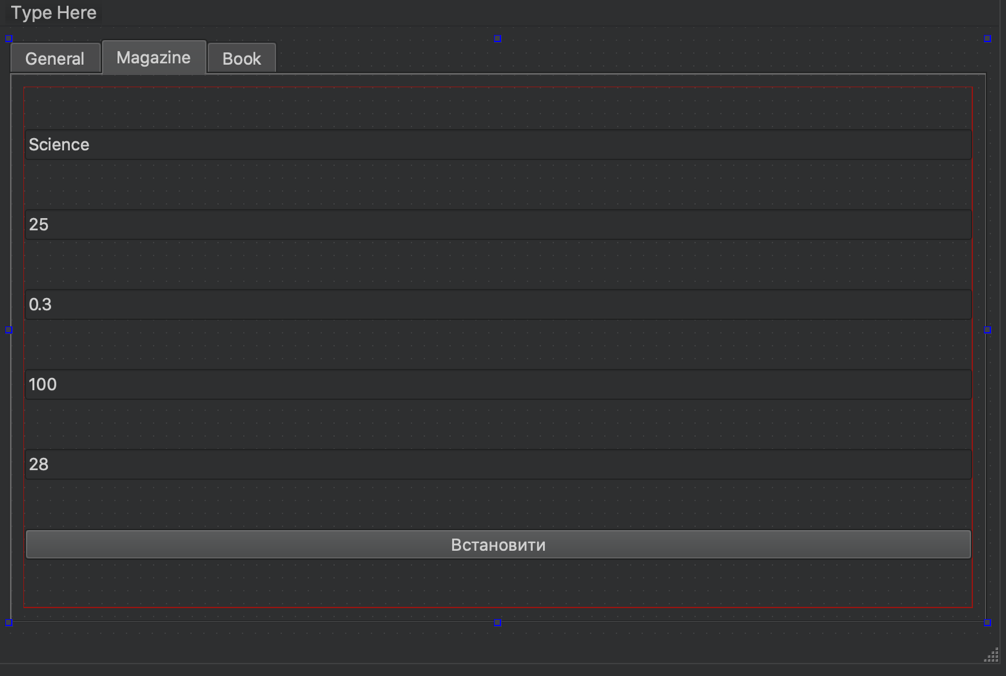
ui->class3\_res->setNum(temp\_book.paper());

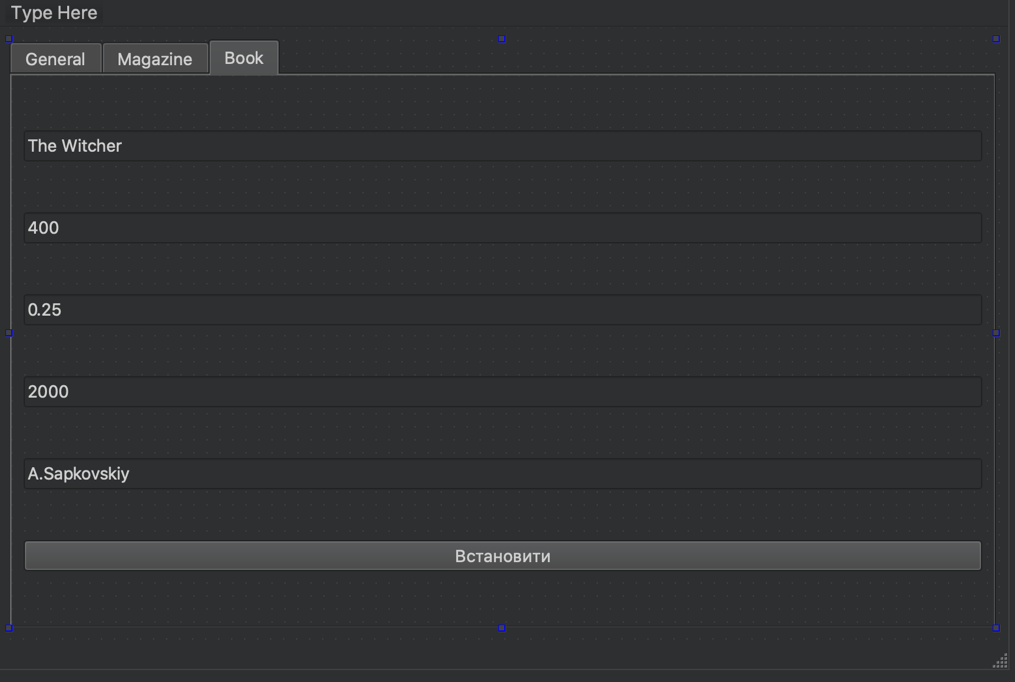
}

Результат виконання:

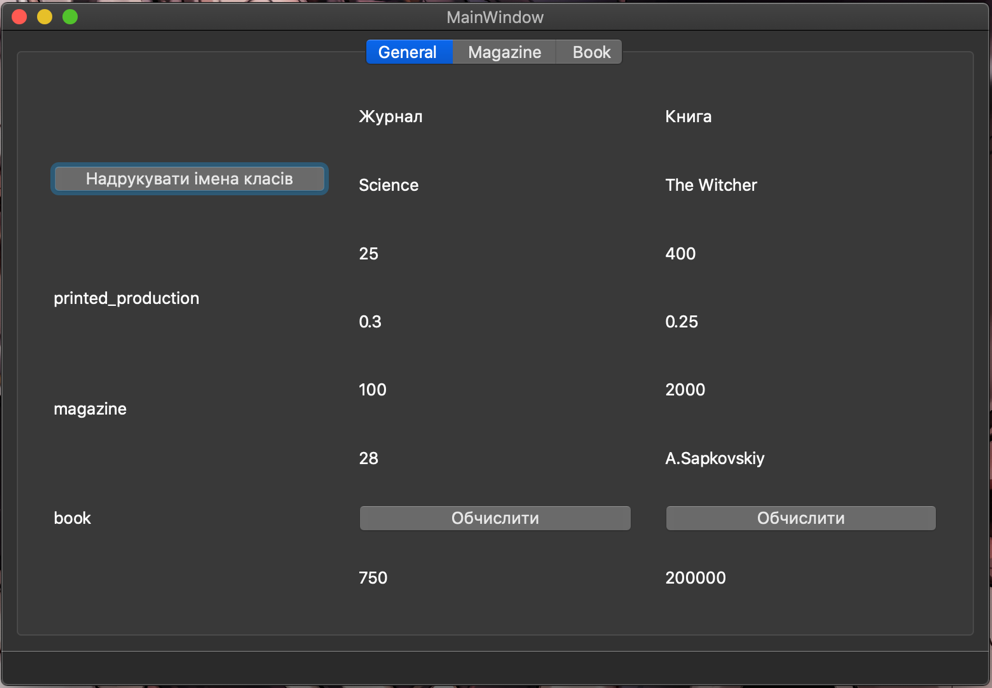
*Форма:*

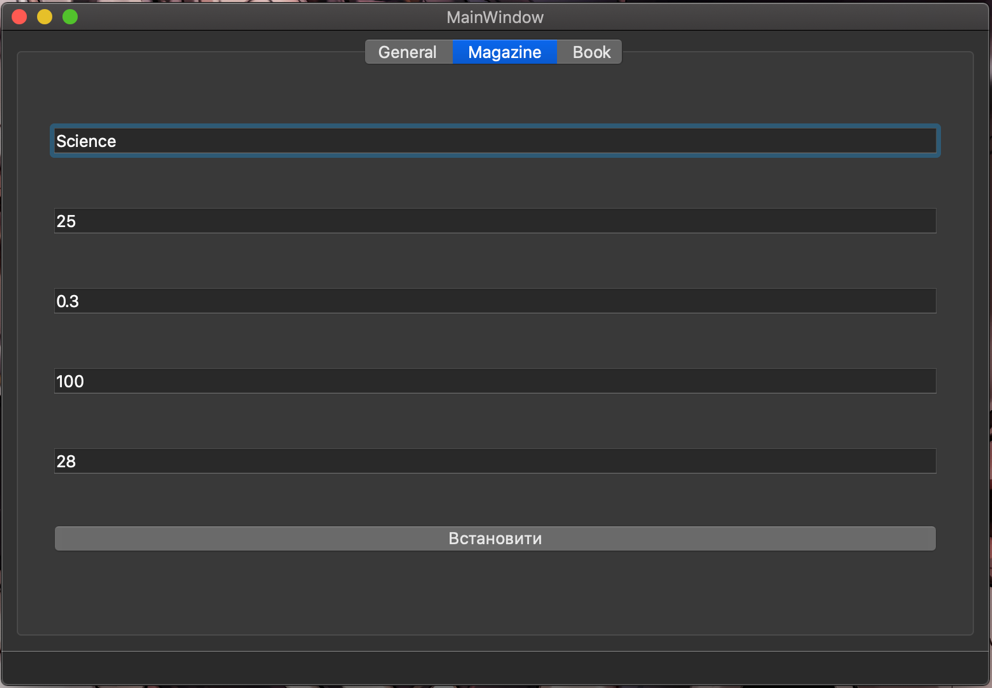
**

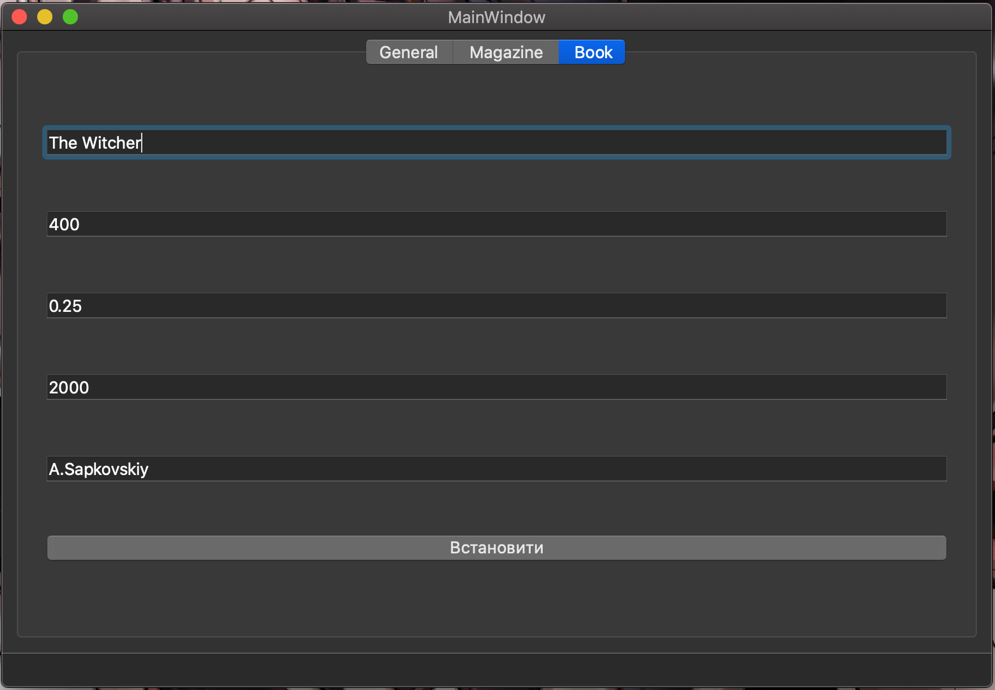
**

**

*Застосунок:*







**Висновки**

Під час виконання лабораторної роботи я навчився створювати базові та похідні класи, використовувати наслідування різного типу доступу, дізнався про можливості та недоліки множинного наслідування. Освоїв принципи та застосування перевизначення методів в похідних класах. Набуті знання закріпив, розробивши графічний застосунок який реалізовує функціонал ієрархії згідно завдання.