Praktyka Programowania

Projekt 1

Tytuł: Księgarnia

Wykonał: Jakub Gałuszka

Grupa: 41M2

1. **Wprowadzenie**

Utworzony program ma na celu nadzór nad książkami kupowanymi przez księgarnię oraz nadzór nad wydatkami i przychodami generowanymi przez księgarnię.

W tym celu utworzono klasę wirtualną, jaką jest budynek. W polu tej klasy znajduje się **cena za wynajem**, istnieje też metoda **Wynajmij()**, ale na potrzeby tego projektu nie będzie używana (założono, że pomieszczenie jest własnością księgarni).

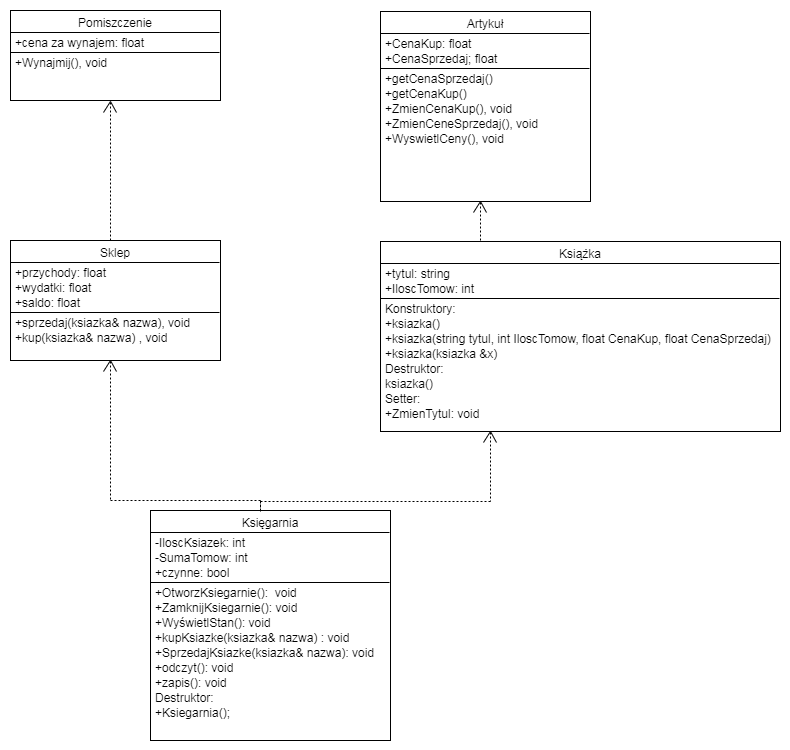
Osobnymi obiektami są **artykuły**. Ich klasa ma zdeklarowaną określoną zmienne: **CenaKup** i **CenaSprzedaj**. Ceny artykułów można zmieniać za pomocą metod **ZmienCeneKup** i **Zmień CeneSprzedaj**. Ceny kupna i sprzedaży można wywołać getterami **getCenaKup()** i **getCenaSprzedaj().** W każdym momencie można wyświetlić aktualne ceny za pomocą metody **WyswietlCeny**. Klasa artykuł również jest klasą wirtualną.

Klasą dziedziczącą z **artykułu** jest **książka**, która poza ceną ma jeszcze **tytul** i określona **IloscTomow.**  Tytuł można zmienić metodą **ZmienTytul(**string nazwa**).** W klasie zdeklarowano destruktor i jego instrukcje. W klasie występują 3 konstruktory: bezparametrowy, wieloparametrowy i kopiujący. Jest to przykład polimorfizmu: Gry funkcja o tej samej nazwie może zostać wywołana na 3 różne sposoby z użyciem różnych argumentów.

Kolejną klasą jest **sklep**, który dziedziczy właściwości z klasy **budynek**. Posiada on pola takie jak **przychody**, **wydatki** i **saldo**. Przychody i wydatki można zmieniać za pomocą metod: **kup(ksiazka& nazwa)** i **sprzedaj(ksiazka& nazwa).** Saldo będzie przeliczane automatycznie poprzez zestawienie przychodów i wydatków.

Klasą dziedziczącą z **Książki** i **Sklepu** jest **Księgarnia**. **Księgarnia** posiada pola takie jak **IloscKsiazek** i **SumaTomow**, które są prywatne. Stan księgarni określa też zmienna **czynne** zmieniana przy użyciu metod **OtworzKsiegarnie()** i **ZamknijKsiegarnie().** Aktualny stan księgarni ilustrujący to czy jest czynna, ilość książek, ilość tomów, przychody, wydatki, saldo można wyświetlić w każdym momencie przy użyciu metody **WyświetlStan**. Książki sprzedaj się i kupuje przy użyciu funkcji **kupKsiazke(ksiazka& nazwa)** i **SprzedajKsiazke(ksiazka& nazwa).** Klasa ma możliwość zapisu do pliku oraz odczytu danych z pliku za pomocą metod **odczyt()** i **zapis()**. Niektórych metod można używać tylko wtedy, gdy księgarnia jest otwarta.

1. **Schemat UML**



1. **Kod źródłowy**

#include <iostream>

#include <fstream>

using namespace std;

**//klasa wirtualna**

class budynek {

public:

float CenaZaWynajem;

**//metoda wirtualna**

void Wynajmij(){

cout << "Budynek ksiegarni wynajeto."<<endl;

}

protected:

private:

};

**//klasa wirtualna**

class artykul {

public:

float CenaKup, CenaSprzedaj;

getCenaKup(){ **//getter 1**

return CenaKup;

}

getCenaSprzedaj(){ **//getter 2**

return CenaSprzedaj;

}

//metoda 1 **//setter**

void ZmienCeneKup(float cena){

this->CenaKup=cena;

cout << "Zmieniono cene kupna artykulu."<<endl;

}

//metoda 2 **//setter**

void ZmienCeneSprzedaj(float cena){

this->CenaSprzedaj=cena;

cout << "Zmieniono cene sprzedazy artykulu."<<endl;

}

//metoda 3 **//setter**

void WyswietlCeny(){

cout << "Cena kupna: " << getCenaKup() <<endl;

cout << "Cena sprzedazy: " << getCenaSprzedaj() << endl;

}

protected:

private:

};

**//klasa standardowa 1**

class ksiazka: public artykul { **//dziedzicznie z klasy artykul**

public:

string tytul;

int IloscTomow;

**//metoda 4**

~ksiazka(){

cout << "Ksiazke: " << this->tytul <<" zniszczono"<< endl;

}

ksiazka(){ **//konstruktor domyslny**

this->tytul="Domyslna";

this->CenaKup=0;

this->CenaSprzedaj=0;

this->IloscTomow=1;

}

**//polimorfizm**

ksiazka(string tytul, int IloscTomow, float CenaKup, float CenaSprzedaj){ **//konstruktor 2**

this->tytul=tytul; **//przyslonienie zmiennych**

this->CenaKup=CenaKup; **//przyslonienie zmiennych**

this->CenaSprzedaj=CenaSprzedaj; **//przyslonienie zmiennych**

this->IloscTomow=IloscTomow; **//przyslonienie zmiennych**

}

**//konstruktor kopiujacy**

ksiazka(ksiazka &x) {

tytul=x.tytul;

IloscTomow=x.IloscTomow;

CenaKup=x.CenaKup;

CenaSprzedaj=x.CenaSprzedaj;

cout << "Stworzono druga taka sama ksiazke."<<endl;

}

**//setter**

void ZmienTytul(string tytul){

this->tytul=tytul;

}

};

**//klasa wirtualna 3**

class sklep: public budynek{ **//dziedziczenie z klasy budynek**

public:

float przychody=0.0, wydatki=0.0, saldo=0.0;

**//metoda**

void sprzedaj(ksiazka& nazwa){

cout<< "Sprzedano artykul." << endl;

this->przychody=this->przychody+nazwa.CenaSprzedaj;

this->saldo=this->saldo+nazwa.CenaSprzedaj;

}

**//metoda**

void kup(ksiazka& nazwa){

cout<< "Zakupiono artykul." << endl;

this->wydatki=this->wydatki+nazwa.CenaKup;

this->saldo=this->saldo-nazwa.CenaKup;

}

};

**//klasa standardowa 2**

class ksiegarnia: public ksiazka, public sklep{

public:

bool czynne=1;

//metoda

void kupKsiazke(ksiazka& nazwa){

if (czynne == 1) {

kup(nazwa);

this->IloscKsiazek=this->IloscKsiazek+1;

this->SumaTomow=this->SumaTomow+nazwa.IloscTomow;

}

else {

cout << "Ksiegarnia jest zamknieta. Nie mozna nic kupic. Otworz ksiegarnie."<< endl;

}

}

**//metoda**

void SprzedajKsiazke(ksiazka& nazwa){

if (czynne == 1) {

sprzedaj(nazwa);

this->IloscKsiazek=this->IloscKsiazek-1;

this->SumaTomow=this->SumaTomow-nazwa.IloscTomow;

}

else{

cout << "Ksiegarnia jest zamknieta. Nie mozna nic sprzedac. Otworz ksiegarnie."<< endl;

}

}

**//setter**

void OtworzKsiegarnie(){

this->czynne=1;

cout<< "Otwarto ksiegarnie"<< endl;

}

**//setter**

void ZamknijKsiegarnie(){

this->czynne=0;

cout<< "Zamknieto ksiegarnie, zapraszamy ponownie."<< endl;

}

**//metoda druku**

void WyswietlStan(){

if (czynne == 1) {

cout << "Ksiegarnia jest czynna."<< endl;

}

else {

cout << "Ksiegarnia jest zamknieta."<< endl;

}

cout << "W ksiegarni jest: " << IloscKsiazek << " ksiazek o lacznej ilosci: " << SumaTomow<< " tomow." << endl;

cout << "Przychody: " << przychody << " Wydatki: " << wydatki << " Saldo: " << saldo <<endl;

}

**//zapis do pliku**

void zapis(){

ofstream plik("plik.txt", ios\_base::app);

plik << IloscKsiazek << " "<< SumaTomow << " " << przychody << " " << wydatki << " " << saldo << endl;

plik.close();

}

**//odczyt z pliku**

void odczyt(){

ifstream plik;

plik.open ("plik.txt");

plik >> IloscKsiazek >> SumaTomow >> przychody >> wydatki >> saldo;

cout << "W ksiegarni jest: " << IloscKsiazek << " ksiazek o lacznej ilosci: " << SumaTomow<< " tomow." << endl;

cout << "Przychody: " << przychody << " Wydatki: " << wydatki << " Saldo: " << saldo << " STAN ODCZYTANO." <<endl;

plik.close();

}

~ksiegarnia(){

cout<<"Zamknieto ksiegarnie"<<endl;

}

private:

int IloscKsiazek=0, SumaTomow=0;

};

int main()

{

ksiazka KA1; **//utworzenie instancji ksiazka (KA1)**

KA1.WyswietlCeny(); **//wywolanie metody druku**

ksiazka KA2("Mama",3,20.00,30.00); **//wywolanie konstruktora 2**

KA2.WyswietlCeny(); **//wywolanie metody druku**

KA1.ZmienCeneKup(20.99); **//wywolanie settera**

KA1.ZmienCeneSprzedaj(21.99); **//wywolanie settera**

KA1.WyswietlCeny(); **//wywolanie metody druku**

ksiazka\* KA3 = new ksiazka(KA2); **//wywolanie konstruktora kopiujacego** z dynamicznym przydzialem pamieci

KA3->tytul = "Mama 2"; **//edycja tytulu (bez settera)**

KA3->ZmienTytul("Mama 3"); **//edycja tytulu (bez settera)**

KA3->WyswietlCeny(); **//wywolanie metody druku**

ksiegarnia\* PI = new ksiegarnia; **//dynamiczne przydzielenie pamieci dla** obkietu typu ksiegarnia

PI->SprzedajKsiazke(KA2); **//metody podczas pracy ksiegarni**

PI->kupKsiazke(KA2);

PI->WyswietlStan();

PI->kupKsiazke(KA2);

PI->WyswietlStan();

PI->ZamknijKsiegarnie();

PI->kupKsiazke(KA2);

PI->zapis(); **//zapis do pliku "plik.txt"**

PI->odczyt(); **//oczyt z pliku "plik.txt"**

delete PI;

delete KA3; **//zwolnienie pamieci**

cout << "KONIEC PROGRAMU." << endl;

return 0;

}