**Простий приклад без конструкторів**

------------------- Rectangle.h ----------------

#pragma once

class Rectangle

{

public:

double a, b; //Поля - властивості

void input();

void print();

double S(); //Функії - методи

double P();

};

--------------- Rectangle.cpp---------------------

#include "stdafx.h"

#include "Rectangle.h"

#include<iostream>

using namespace std;

void Rectangle::input()

{

printf("a=");

cin >> a;

printf("b=");

cin >> b;

}

void Rectangle::print()

{

printf("a=%f , b=%f",a,b);

}

double Rectangle::S()

{

return a\*b;

}

double Rectangle::P()

{

return 2\*(a+b);

}

--------------------------- main-----------------------

// ConsoleApplication28.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include "Rectangle.h"

#include<iostream>

using namespace std;

int main()

{

Rectangle\* rectangles = new Rectangle[3];

for (int i = 0; i < 3; i++)

{

rectangles[i].input();

}

double total\_S = 0;

for (int i = 0; i < 3; i++)

{

total\_S += rectangles[i].S();

}

printf("S=%f",total\_S);

return 0;

}

==================================

**Приклад з прямокутником з конструкторами**

#pragma once

class Rectangle

{

public:

//1.Поля

double a, b; //Поля - властивості

//------------------------

//2.Конструктори

Rectangle(double a, double b); //Конструтор з параметрами

Rectangle(double a); //Конструтор з параметрами

Rectangle(); //Конструтор за замовчуванням

Rectangle(const Rectangle& obj); //Конструктор копіювання

//3.Інші функції-методи

void input();

void print();

double S(); //Функії - методи

double P();

};

=================================================================

#include "stdafx.h"

#include "Rectangle.h"

#include<iostream>

using namespace std;

Rectangle::Rectangle(double a, double b)

{

this->a = a;

this->b = b;

}

Rectangle::Rectangle(double a):Rectangle(a,a)

{}

Rectangle::Rectangle():Rectangle(0,0)

{

}

Rectangle::Rectangle(const Rectangle & obj):Rectangle(obj.a,obj.b)

{}

void Rectangle::input()

{

printf("a=");

cin >> a;

printf("b=");

cin >> b;

}

void Rectangle::print()

{

printf("a=%f , b=%f",a,b);

}

double Rectangle::S()

{

return a\*b;

}

double Rectangle::P()

{

return 2\*(a+b);

}

====================================================

// ConsoleApplication28.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include "Rectangle.h"

#include<iostream>

using namespace std;

int main()

{

Rectangle\* rectangles = new Rectangle[3];

for (int i = 0; i < 3; i++)

{

rectangles[i].input();

}

double total\_S = 0;

for (int i = 0; i < 3; i++)

{

total\_S += rectangles[i].S();

}

printf("S=%f",total\_S);

return 0;

}

=======

Задача. Описати клієнта банку (ім’я, рахунок, баланс, кредитний ліміт). Методи: вводу, виводу, додваня грошей, вилучення грошей, визначення максимальної суми, яка може бути знята.

----------------------- Client.h--------------------------

//1.Опис полів

string Name;

int AccountNumber;

double Balance;

double CreditLimit;

//2.Опис конструткорів

Client(string name, int accountNumber, double balance, double creditLimit);

Client(string name, int accountNumber, double balance);

Client();

//3.Опис інших функцій

void input();

void print();

void putMoney(double money);

bool withdrawMoney(double money);

double getMaxSum();

----------------- Client.cpp ------------------------------

#include "stdafx.h"

#include "Client.h"

#include<iostream>

using namespace std;

Client::Client(string name, int accountNumber, double balance, double creditLimit)

{

Name = name;

AccountNumber = accountNumber;

Balance = balance;

CreditLimit = creditLimit;

}

Client::Client(string name, int accountNumber, double balance):Client(name,accountNumber,balance,3000)

{

}

Client::Client():Client("no name",0,0,0)

{

}

void Client::input()

{

printf("Name : ");

cin >> Name;

printf("Account number : ");

cin >> AccountNumber;

printf("Balance : ");

cin >> Balance;

printf("Credit : ");

cin >> CreditLimit;

}

void Client::print()

{

printf("%s: balance:%f, credit:%f\n", Name.data(),Balance,CreditLimit);

}

void Client::putMoney(double money)

{

Balance += money;

}

bool Client::withdrawMoney(double money)

{

if (money<=getMaxSum())

{

Balance -= money;

return true;

}

return false;

}

double Client::getMaxSum()

{

return Balance+CreditLimit;

}

--------------------- main ---------------------------------

// ConsoleApplication29.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

#include "Client.h"

using namespace std;

int main()

{

Client\* client = new Client();

client->input();

int answer;

double money;

do {

system("cls");

printf("1.Print info\n");

printf("2.Put money \n");

printf("3.Withdraw money \n");

printf("4.Exit\n");

printf("Your choise:\n");

cin >> answer;

switch (answer)

{

case 1: client->print();

break;

case 2:printf("Money=");

cin >> money;

client->putMoney(money);

break;

case 3:printf("Money=");

cin >> money;

if (client->withdrawMoney(money))

{

printf("Take your money");

}

else

{

printf("Not enought on your balance");

}

break;

}

system("pause");

} while (answer!=4);

delete client;

system("pause");

return 0;

}

------------------- main із додавання і вилученням клієнтів (погано, бо все тут і у масиві. Краще у окремому класі і зберігати у файлі або у базі даних !!!! )---------------------------------------------------

// ConsoleApplication29.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

#include "Client.h"

using namespace std;

int main()

{

Client\* clients = new Client[100];

int clientsCount = 0;

int answer;

double money;

do {

system("cls");

printf("1.Print info\n");

printf("2.Add client \n");

printf("3.Delete client \n");

printf("4.Exit\n");

printf("Your choise:\n");

cin >> answer;

switch (answer)

{

case 1:

for (int i = 0; i < clientsCount; i++)

{

clients[i].print();

}

break;

case 2:

clientsCount++;

clients[clientsCount - 1].input();

break;

case 3:printf("Account number=");

int accountNumber;

cin >> accountNumber;

int clientIndex = -1;

for (int i = 0; i < clientsCount; i++)

{

if (clients[i].AccountNumber == accountNumber)

{

clientIndex = i;

break;

}

}

if (clientIndex==-1)

{

printf("No");

}

else

{

clients[clientIndex] = clients[clientsCount - 1];

clientsCount--;

}

break;

}

system("pause");

} while (answer!=4);

delete[] client;

system("pause");

return 0;

}