КУРС: «ПЛАТФОРМА MICROSOFT .NET ТА МОВА ПРОГРАМУВАННЯ C#

Тема: Простір імен. Індексатори.

**Модуль 3. Частина 1**

**Завдання 1**

Реалізуйте клас «Співробітник». Необхідно зберігати у полях класу: ПІБ, дату народження, контактний телефон, робочий email, посаду, опис службових обов’язків. Реалізуйте методи класу для введення даних, виведення даних. Реалізуйте доступ до полів через властивості класу.

Додайте до вже створеного класу інформацію про заробітну плату працівника. Виконайте перезавантаження + (для збільшення зарплати на вказану кількість), – (для зменшення зарплати на вказану кількість), == (перевірка на рівність зарплат працівників), < і > (перевірка на меншу чи більшу кількість зарплат працівників), != і ==. Використовуйте механізм властивостей полів класу.

**Завдання 2**

Написати додаток, що імітує роботу банкомату

Реалізувати класи **Banc**, **Client**, **Account** у різних просторах імен (загальний простір імен **Bankomat**). Спочатку клієнту потрібно відкрити рахунок у банку, отримати номер рахунку, отримати свій пароль, покласти суму на рахунок.

1. Додаток пропонує ввести пароль передбачуваної кредитної картки, надається 3 спроби на правильне введення пароля. Якщо спроби вичерпані, програма видає відповідне повідомлення та завершується.

2. При успішному введенні пароля відображається меню. Користувач може вибрати одну з кількох дій:

- Виведення балансу на екран

- поповнення рахунку

- зняти гроші з рахунку

- Вихід

3. Якщо користувач вибирає виведення балансу на екран, програма відображає стан передбачуваного рахунку, після чого пропонує або повернутися до меню, або зробити вихід.

4. Якщо користувач вибирає поповнення рахунку, програма запитує суму для поповнення та виконує операцію, супроводжуючи її висновком відповідного коментаря. Потім потрібно пропозицію повернутися в меню або виконати вихід.

5. Якщо користувач вибирає зняти гроші з рахунку, програма запитує суму. Якщо сума перевищує суму рахунку користувача, програма видає повідомлення та переказує користувача в меню. Інакше відображає повідомлення про те, що суму знято з рахунку та зменшує суму рахунку на вказану величину.

Завдання 1

using System;

namespace test11

{

class Employee

{

string pib;

DateTime datetime;

string tel;

string email;

string position;

string responsibilities;

float salary;

public string PIB

{

get { return this.pib; }

set { this.pib = value; }

}

public DateTime DATEtime

{

get { return this.datetime; }

set { this.datetime = value; }

}

public string Tel

{

get { return this.tel; }

set { this.tel = value; }

}

public string Email

{

get { return this.email; }

set { this.email = value; }

}

public string Position

{

get { return this.position; }

set { this.position = value; }

}

public string Responsibilities

{

get { return this.responsibilities; }

set { this.responsibilities = value; }

}

public float Salary

{

get { return this.salary; }

set { this.salary = value; }

}

public Employee()

{

this.pib = "Unknown";

this.datetime = new DateTime(1, 1, 1);

this.tel = "Unknown";

this.email = "Unknown";

this.position = "Unknown";

this.responsibilities = "Unknown";

this.salary = 0;

}

public Employee(string pib, DateTime datetime, string tel, string email, string position, string responsibilities, float salary)

{

this.pib = pib;

this.datetime = datetime;

this.tel = tel;

this.email = email;

this.position = position;

this.responsibilities = responsibilities;

this.salary = salary;

}

public void SetParameters(string pib, DateTime datetime, string tel, string email, string position, string responsibilities, float salary)

{

this.pib = pib;

this.datetime = datetime;

this.tel = tel;

this.email = email;

this.position = position;

this.responsibilities = responsibilities;

this.salary = salary;

}

public void PrintInfo()

{

Console.WriteLine(pib);

Console.WriteLine("Дата народження: " + datetime.ToShortDateString());

Console.WriteLine("Номер телефону: " + tel);

Console.WriteLine("Пошта: " + email);

Console.WriteLine("Посада: " + position);

Console.WriteLine("Обов'язки:\n" + responsibilities);

Console.WriteLine("Зарплата: " + salary + "$");

}

public static Employee operator +(Employee employee, float increase)

{

employee.salary += increase;

return employee;

}

public static Employee operator +(float increase, Employee employee)

{

employee.salary += increase;

return employee;

}

public static Employee operator -(Employee employee, float decrease)

{

if (decrease < employee.salary)

employee.salary -= decrease;

return employee;

}

public static bool operator ==(Employee employee1, Employee employee2)

{

return employee1.salary == employee2.salary;

}

public static bool operator !=(Employee employee1, Employee employee2)

{

return employee1.salary != employee2.salary;

}

public static bool operator >(Employee employee1, Employee employee2)

{

return employee1.salary > employee2.salary;

}

public static bool operator <(Employee employee1, Employee employee2)

{

return employee1.salary < employee2.salary;

}

public override string ToString()

{

return (pib + "\nДата народження: " + datetime.ToShortDateString() + "\nНомер телефону: " + tel + "\nПошта: " + email + "\nПосада: " + position + "\nОбов'язки:\n" + responsibilities + "\nЗарплата: " + salary.ToString() + "$");

}

}

class Program

{

public static void Main()

{

System.Console.OutputEncoding = System.Text.Encoding.UTF8;

Employee e1 = new Employee();

e1.Salary = 1200;

e1.DATEtime = new DateTime(2004, 12, 1);

Employee e2 = new Employee();

e2.Salary = 1200;

e2.DATEtime = new DateTime(2006, 4, 12);

bool b = e1 < e2;

Console.WriteLine(e1);

}

}

}

Завдання 2

using System;

namespace NSBanc

{

using NSClient;

class Bank

{

Client[] clients;

int endindex;

int[] numbers;

float[] balance;

public Bank(int size)

{

endindex = 0;

clients = new Client[size];

numbers = new int[size];

balance = new float[size];

}

bool CorrectPassword(int ind,string pass\_word)

{

if (string.Equals(clients[ind].ACcount.Password, pass\_word) == true)

return true;

return false;

}

bool IsOrNoIs(int num)

{

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

{

if (num == numbers[i])

return true;

}

return false;

}

public int RegistrationNewClient(Client client)

{

clients[endindex] = client;

int num = (new Random()).Next(1, 100000);

while (IsOrNoIs(num))

{

num = (new Random()).Next(1, 100000);

}

numbers[endindex] = num;

endindex++;

return num;

}

public void LogIn(int number\_account)

{

int index = -1;

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

{

if(numbers[i]== number\_account)

{

index = i;

break;

}

}

if(index!=-1)

{

bool b = false;

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

{

Console.Write("Введіть пароль(3 спроби)->");

string pass\_word = Console.ReadLine();

if (CorrectPassword(index, pass\_word) == true)

{

b = true;

break;

}

}

if(b==true)

{

float suma;

int k = 1;

while (k != 0)

{

Console.WriteLine("Показати баланс ->1\nПоповнити рахунок ->2\nЗняти гроші з рахунку ->3\nВихід ->0");

Console.Write("->");

k = Convert.ToInt32(Console.ReadLine());

Console.Clear();

switch (k)

{

case 1:

Console.WriteLine("Ваш баланс: "+balance[index]+"грн");

break;

case 2:

Console.Write("Введіть суму на яку поповнити ->");

suma = Convert.ToInt32(Console.ReadLine());

balance[index] += suma;

break;

case 3:

Console.Write("Введіть суму яку хочете зняти ->");

suma = Convert.ToInt32(Console.ReadLine());

if(suma>balance[index])

{

Console.WriteLine("Недостатньо коштів");

}

else

{

balance[index] -= suma;

Console.WriteLine("Успішно знято");

}

break;

}

}

}

else

{

Console.WriteLine("Ви вичерпали спроби ввести пароль");

}

}

}

}

}

namespace NSClient

{

using NSAccount;

class Client

{

string pib;

int age;

string passportdata;

Account account;

public Client(string pib, int age, string passportdata, Account account)

{

this.pib = pib;

this.age = age;

this.passportdata = passportdata;

this.account = account;

}

public string PIB

{

get { return pib; }

set { pib = value; }

}

public int Age

{

get { return age; }

set { age = value; }

}

public string Passportdata

{

get { return passportdata; }

set { passportdata = value; }

}

public Account ACcount

{

get { return account; }

set { account = value; }

}

public void PrintInfo()

{

Console.WriteLine(pib);

Console.WriteLine("Вiк: "+age+"р.");

Console.WriteLine("Паспортнi данi: " + passportdata);

account.PrintInfo();

}

}

}

namespace NSAccount

{

class Account

{

string login;

string password;

public Account()

{

this.login = "Unknown";

this.password = "Unknown";

}

public Account(string login, string password)

{

this.login = login;

this.password = password;

}

public string Login

{

get { return login; }

set { login = value; }

}

public string Password

{

get { return password; }

set { password = value; }

}

public void PrintInfo()

{

Console.WriteLine("Логiн: " + login + " Пароль: " + password);

}

}

}

namespace Bankomat

{

using NSAccount;

using NSBanc;

using NSClient;

class Program

{

public static void Main()

{

System.Console.OutputEncoding = System.Text.Encoding.UTF8;

Bank bank = new Bank(5);

int card1= bank.RegistrationNewClient(new Client("Lashch Anton Vadymovych", 18, "AN-23-24-JK", new Account("antonlashch@icloud.com", "787898")));//видає "картку" її номер

int card2 = bank.RegistrationNewClient(new Client("Derkach Ivan Ruslanovych", 19, "AB-13-49-lK", new Account("ivanderkach@icloud.com", "123456")));

bank.LogIn(card1);

bank.LogIn(card2);

}

}

}