Лабораторная работа №8

Упражнение 1: Создание перечисления.

В этом упражнении Вы создадите перечисление для представления различных типов банковских счетов. Затем Вы используете это перечисление для создания двух переменных, которым Вы присвоите значения Checking и Deposit. Далее Вы выведете на экран значения этих переменных, используя функцию System.Console.WriteLine.

Создайте перечисление

Создайте проект BankAccount.sln в папке install folder\Labs\Lab03\Starter\BankAccount.

Переменуйте файл Program.cs на файл Enum.cs, согласитесь с предложением изменить ссылки на новое имя.

Перед описанием класса добавьте перечисление AccountType:

public enum AccountType { Checking, Deposit }

Данное перечисление содержит типы Checking и Deposit.

В методе Main объявите две переменные типа AccountType:

AccountType goldAccount;

AccountType platinumAccount;

Присвойте первой переменной значение Checking, а второй - Deposit: goldAccount = AccountType.Checking; platinumAccount = AccountType.Deposit;

Выведите на консоль значения обеих переменных, два раза используя метод Console.WriteLine:

Console.WriteLine("The Customer Account Type is {0}", goldAccount);

Console.WriteLine("The Customer Account Type is {0}", platinumAccount);

Откомпилируйте и запустите программу.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BankAccount

{

public enum AccountType

{

Checking,

Deposit

}

class Enum

{

static void Main(string[] args)

{

AccountType goldAccount;

AccountType platinumAccount;

goldAccount = AccountType.Checking;

platinumAccount = AccountType.Deposit;

Console.WriteLine("The Customer Account Type is {0}", goldAccount);

Console.WriteLine("The Customer Account Type is {0}", platinumAccount);

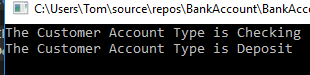
Console.ReadKey();

}

}

}

Скриншоты выполнения:



Упражнение 2: Создание и использование структуры

В этом упражнении Вы создадите структуру, которую можно использовать для представления банковских счетов. Для хранения номеров счетов (тип данных long), балансов счетов (тип данных decimal) и типов счетов (перечисление, созданное в упражнении 1) будете использовать переменные. Затем создадите переменную типа структуры, заполните ее данными и выведете результаты на консоль.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BankAccount

{

public enum AccountType

{

Checking,

Deposit

}

public struct BankAccount

{

public long accNo;

public decimal accBal;

public AccountType accType;

}

class Enum

{

static void Main(string[] args)

{

BankAccount goldAccount;

BankAccount platinumAccount;

goldAccount.accType = AccountType.Checking;

goldAccount.accBal = (decimal)3200.00;

goldAccount.accNo = 123;

Console.WriteLine("\*\*\* Account Summary \*\*\*");

Console.WriteLine("Acct Number {0}", goldAccount.accNo);

Console.WriteLine("Acct Type {0}", goldAccount.accType);

Console.WriteLine("Acct Balance ${0}", goldAccount.accBal);

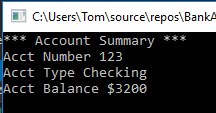
Console.ReadKey();

}

}

}

Скриншоты выполнения:



Упражнение 3: Добавление возможности ввода/вывода

В этом упражнении Вы измените код, написанный в упражнении 2. Вместо использования счета номер 123, Вы будете запрашивать номер счета у пользователя, а потом использовать его при выводе информации о банковском счете на консоль.

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BankAccount

{

public enum AccountType

{

Checking,

Deposit

}

public struct BankAccount

{

public long accNo;

public decimal accBal;

public AccountType accType;

}

class Enum

{

static void Main(string[] args)

{

BankAccount goldAccount;

BankAccount platinumAccount;

goldAccount.accType = AccountType.Checking;

goldAccount.accBal = (decimal)3200.00;

Console.Write("Enter account number: ");

goldAccount.accNo = long.Parse(Console.ReadLine());

Console.WriteLine("\*\*\* Account Summary \*\*\*");

Console.WriteLine("Acct Number {0}", goldAccount.accNo);

Console.WriteLine("Acct Type {0}", goldAccount.accType);

Console.WriteLine("Acct Balance ${0}", goldAccount.accBal);

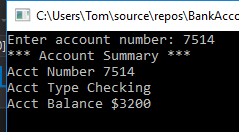
Console.ReadKey();

}

}

}

Скриншоты выполнения:



Задание: написать программу согласно заданию с использованием структуры.

|  |  |  |
| --- | --- | --- |
| 11 | Наименование, Цена, Дата производства, Срок годности, Количество, Производитель | Вывести все сведения о просроченных товарах. |

Решение:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Collections;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

namespace ConsoleApp3

{

struct Product

{

public string Name;

public int Price;

public DateTime Manufactured;

public DateTime Expires;

public int Quantity;

public string Company;

public Product(string Name, int Price, DateTime Manufactured,

DateTime Expires, int Quantity, string Company)

{

this.Name = Name;

this.Price = Price;

this.Manufactured = Manufactured;

this.Expires = Expires;

this.Quantity = Quantity;

this.Company = Company;

}

}

class Program

{

static void Main()

{

try

{

Console.Write("Кол-во товаров: ");

int n = int.Parse(Console.ReadLine());

Product[] Products = new Product[n];

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

{

Console.WriteLine("Заполнение данных {0} товара.", i + 1);

Console.Write("Наименование: ");

string Name = Console.ReadLine();

Console.Write("Цена: ");

int Price = Int32.Parse(Console.ReadLine());

Console.Write("Дата изготовления: ");

DateTime Manufactured = DateTime.Parse(Console.ReadLine());

Console.Write("Срок годности: ");

DateTime Expires = DateTime.Parse(Console.ReadLine());

Console.Write("Количество: ");

int Quantity = Int32.Parse(Console.ReadLine());

Console.Write("Производитель: ");

string Company = Console.ReadLine();

Console.WriteLine();

Products[i] = new Product(Name, Price, Manufactured, Expires, Quantity, Company);

}

Console.WriteLine("Информация о просроченных товарах:");

foreach (Product a in Products)

{

if (a.Expires < DateTime.Today)

{

Console.WriteLine("Наименование: {0}", a.Name);

Console.WriteLine("Цена:{0}", a.Price);

Console.WriteLine("Дата изготовления: {0}", a.Manufactured);

Console.WriteLine("Срок годности: {0}", a.Expires);

Console.WriteLine("Количество: {0}", a.Quantity);

Console.WriteLine("Производитель: {0}", a.Company);

Console.WriteLine();

}

}

}

catch (Exception e)

{

Console.WriteLine(e.ToString());

}

finally

{

Console.ReadKey();

}

}

}

}

Тестирование:

