Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра Информатики

Дисциплина «Программирование»

**ОТЧЕТ**

к лабораторной работе №4

на тему:

**«Конструкторы. статические члены класса. шаблон проектирования singleton»**

БГУИР 6-05-0612-02 35

|  |
| --- |
| Выполнил студент группы 353504  ГУСЕНЦОВА Екатерина Андреевна |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 Индивидуальное задание

**Задание 1. Вариант 5.** Предметная область: Банк-Вклад. В классе хранить информацию о наименовании банка, число вкладов, размер вклада (одинаковый для всех вкладчиков - класс Вклад), размер процентной ставки. Реализовать метод для подсчета общей выплаты по процентам. Реализовать возможность изменения (увеличения и уменьшения) размера вклада.

# 2 Выполнение работы

Код, представленный ниже является простую консольную программу для работы с банковскими вкладами.

using System;  
  
namespace BankDeposit  
{  
 internal class Program  
 {  
 static void Main(string[] args)  
 {  
 bool continueTheProgram = false;  
 do  
 {  
 Bank bank = Bank.GetInstance();  
 Console.Write("Введите наименование банка: ");  
 bank.SetBankName(Console.ReadLine());  
 Console.Write("Введите число вкладов: ");  
 bank.SetNumberOfDeposits(CheckingForInput.Int(Conversion.IntToString(bank.GetNumberOfDeposits())));  
 Console.Write("Введите размер вклада: ");  
 bank.SetDepositAmount(CheckingForInput.Double(Conversion.DoubleToString(bank.GetDepositAmount())));  
 Console.Write("Введите размер процентной ставки: ");  
 bank.SetInterestRate(CheckingForInput.Percent(Conversion.IntToString(bank.GetInterestRate())));  
 double totalInterest = bank.CalculateTotalInterest();  
 Console.WriteLine("\tОбщая выплата по процентам составляет " + totalInterest + ".");  
 Console.Write("\nХотите изменить размер вклада? (1 - Да, 2 - Нет): ");  
 string str = "\0";  
 str = CheckingForInput.ContinuationAndTermination(str);  
 switch (str)  
 {  
 case "1":  
 Console.Write("\nВведите новый размер вклада: ");  
 double newAmount = CheckingForInput.Double(Conversion.DoubleToString(bank.GetDepositAmount()));  
 bank.ChangeDepositAmount(newAmount);  
 totalInterest = bank.CalculateTotalInterest(); // Пересчитываем общую выплату после изменения размера вклада  
 Console.WriteLine("\tНовая общая выплата по процентам составляет " + totalInterest + ".");  
 break;  
 case "2":  
 break;  
 }  
 Console.Write("\nХотите продолжить? (1 - Да, 2 - Нет): ");

str = "\0";  
 str = CheckingForInput.ContinuationAndTermination(str);  
 switch (str)  
 {  
 case "1":  
 continueTheProgram = true;  
 break;  
 case "2":  
 continueTheProgram = false;  
 break;  
 }  
 } while (continueTheProgram);  
 }  
 }  
  
 internal class Bank  
 {  
 // Fields  
 private static Bank instance;  
 private string bankName;  
 private int numberOfDeposits;  
 private Deposit deposit;  
 private int interestRate;  
  
 // Properties   
 public string GetBankName() => bankName;  
 public void SetBankName(string name) => bankName = name;  
 public int GetNumberOfDeposits() => numberOfDeposits;  
 public void SetNumberOfDeposits(int count) => numberOfDeposits = count;  
 public double GetDepositAmount() => deposit.Amount;  
 public void SetDepositAmount(double amount) => deposit.Amount = amount;  
 public int GetInterestRate() => interestRate;  
 public void SetInterestRate(int rate) => interestRate = rate;  
  
 // Methods  
 private Bank()  
 {  
 deposit = new Deposit();  
 }  
 public static Bank GetInstance()  
 {  
 if (instance == null)  
 {  
 instance = new Bank();  
 }  
 return instance;  
 }  
  
 public double CalculateTotalInterest()  
 {  
 double totalInterest = deposit.Amount \* ((double)interestRate / 100) \* numberOfDeposits;  
 return totalInterest;  
 }  
  
 public void ChangeDepositAmount(double newAmount)  
 {  
 deposit.Amount = newAmount;  
 }  
 }  
  
 internal class Deposit  
 {  
 public double Amount { get; set; }  
 }  
  
 internal class CheckingForInput  
 {  
 static public string ContinuationAndTermination(string s)  
 {  
 while (true)  
 {  
 s = Console.ReadLine();  
 if (s == "1" || s == "2")  
 {  
 return s;  
 }  
 else  
 {  
 Console.Write("Неправильный ввод! Попробуйте еще раз: ");  
 }  
 }  
 }  
  
 static public int Int(string s)  
 {  
 int output;  
 while (true)  
 {  
 s = Console.ReadLine();  
 if (int.TryParse(s, out output) && output > 0)  
 {  
 return output;  
 }  
 else  
 {  
 Console.Write("Неправильный ввод! Попробуй еще раз: ");  
 }  
 }  
 }  
  
 static public double Double(string s)  
 {  
 double output;  
 while (true)  
 {  
 s = Console.ReadLine();  
 if (double.TryParse(s, out output) && output > 0)  
 {  
 return output;  
 }  
 else  
 {  
 Console.Write("Неправильный ввод! Попробуй еще раз: ");  
 }  
 }  
 }  
  
 static public int Percent(string s)  
 {  
 int output;  
 while (true)  
 {  
 s = Console.ReadLine();  
 if (int.TryParse(s, out output) && output >= 0 && output <= 100)  
 {  
 return output;  
 }  
 else  
 {  
 Console.Write("Неправильный ввод! Попробуй еще раз: ");  
 }  
 }  
 }  
 }  
  
 internal class Conversion  
 {  
 static public string IntToString(int value) => value.ToString();  
 static public string DoubleToString(double value) => value.ToString();  
 }  
}

Вот пошаговое объяснение того, что происходит в коде:

Класс Program содержит метод Main, который является точкой входа в программу. Внутри метода Main осуществляется основной поток выполнения программы. Класс Bank представляет банк и хранит информацию о банке, такую как его наименование, количество вкладов, размер вклада и процентную ставку. Он также содержит методы для работы с этой информацией, такие как установка и получение значений для каждого атрибута, а также методы для расчета общей выплаты по процентам и изменения размера вклада. Класс Deposit представляет вклад и содержит только одно свойство - Amount, которое представляет собой сумму вклада. Класс CheckingForInput содержит статические методы для проверки ввода данных пользователя. Он используется для проверки правильности ввода чисел и процентных ставок. Класс Conversion содержит статические методы для преобразования целых чисел и чисел с плавающей точкой в строки. Он используется для преобразования значений перед выводом их в консоль.

Основная логика программы в методе Main: В методе Main программа входит в цикл do-while, который позволяет пользователю продолжать выполнение программы или завершить её. Внутри цикла сначала запрашивается информация о банке и вкладах, а затем вычисляется общая выплата по процентам. После этого пользователю предлагается изменить размер вклада. Если пользователь соглашается, размер вклада изменяется, а затем пересчитывается общая выплата по процентам. После каждой операции пользователю предлагается продолжить или завершить программу.

Этот код создает простую консольную программу для работы с банковскими вкладами, позволяя пользователю управлять информацией о банке и вкладах.

Результат выполнения индивидуального задания представлен на рисунке 1.

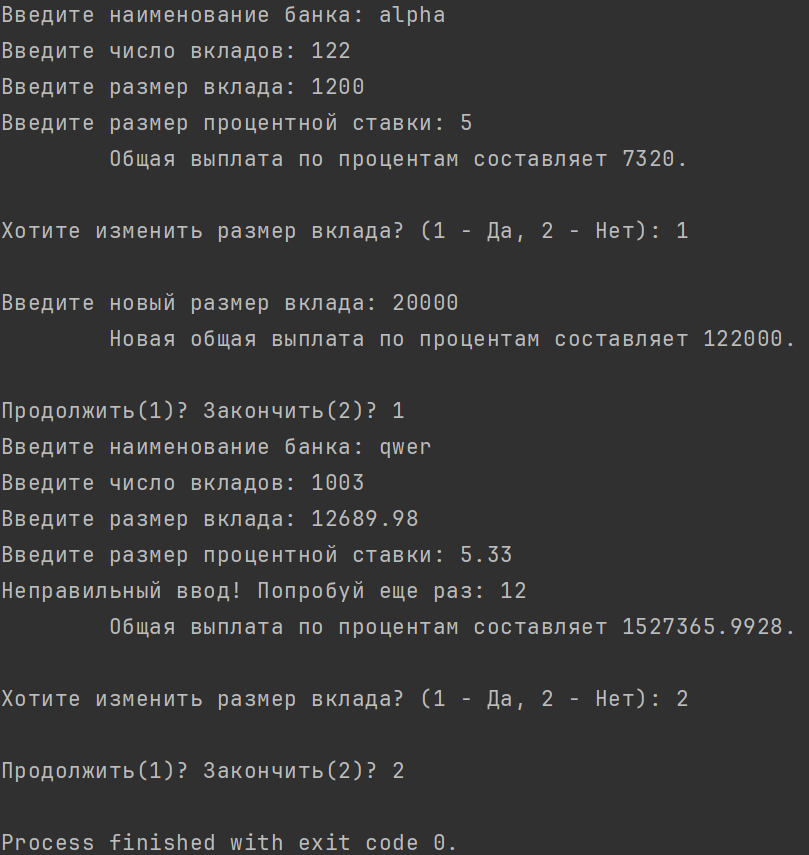
=

Рисунок 1 – Результат выполнения программы

# Вывод

В ходе лабораторной работы было изучено назначение конструкторов и их использование, освоены все возможности членов класса с модификатором static. Был изучен шаблон проектирования Singleton.