**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4**

**Тема:** Процедуры и функции. Перегрузка функций (методов) и операторов

**Цель:** Разрабатывать функции. Реализовывать перегрузку функций и операторов. Использовать функции и операторы

**Время выполнения**: 6 часов

**1 Входной контроль**

1. Опишите, как выполняется перегрузка методов в C#.

Иногда возникает необходимость создать один и тот же метод, но с разным набором параметров. И в зависимости от имеющихся параметров применять определенную версию метода. Такая возможность еще называется перегрузкой методов

1. Опишите, как выполняется перегрузка операторов в C#.

Для перегрузки оператора служит ключевое слово operator, определяющее операторный метод, который, в свою очередь, определяет действие оператора относительно своего класса. Существуют две формы операторных методов (operator): одна - для унарных операторов, другая - для бинарных.

1. Опишите отличие унарных и бинарных операторов.

**Унарные**. Работают с одним операндом.

**Бинарные**. Работают с двумя операндами (левым и правым).

**2 Задания для выполнения работы 2.1**

**Задание № 1. Разработать класс согласно заданию.**

Разработать класс «Счет», который имеет следующие атрибуты: сумма и валюта счета. Реализовать в классе методы для просмотра и установки атрибутов класса, реализовать конструкторы, метод для ввода значений всех атрибутов с клавиатуры и метод для вывода всех значений атрибутов в консоль. Реализовать перегрузку следующих операторов: + (слияние счетов); - = (снятие денег со счета); += (пополнение счета); > (сравнение двух счетов); ++ (начисление процентов на текущую сумму на счете (необходимо определить размер начисляемых процентов (5%) и добавить к текущей сумме)). Примечание: необходимо учитывать, что операции со счетами возможны только в том случае, если совпадает валюта счетов

|  |  |
| --- | --- |
| Класс | Описание |
| Счет | Сумма (double), Валюта(строка). |

Реализация класса «Счет» представлена в листинге 1.

Листинг 1 – Класс Счет

**using** System;

**using** System.Collections.Generic;

**using** System.Linq;

**using** System.Text;

**using** System.Threading.Tasks;

**namespace** Laba4

{

**class** Schet

{

**public** **double** Summa { get; set; }

**public** **string** Value { get; set; }

**public** Schet()

{

Summa = 0;

Value = null;

}

**public** Schet(**double** summa, **string** value)

{

Summa = summa;

Value = value;

}

**public** **void** Input()

{

Console.Write("Введите сумму: ");

Summa = Convert.ToDouble(Console.ReadLine());

Console.Write("Введите валюту: ");

Value = Console.ReadLine();

}

**public** **void** Print()

{

Console.WriteLine("Сумма: {0}; Валюта: {1}.", Summa, Value);

}

**public** **static** Schet operator +(Schet schet1, Schet schet2)

{

**return** **new** Schet { Summa = schet1.Summa + schet2.Summa, Value =

schet1.Value };

}

**public** **static** Schet operator -(Schet schet1, **double** x)

{

**return** **new** Schet { Summa = schet1.Summa - x, Value = schet1.Value};

}

**public** **static** Schet operator +(Schet schet1, **double** x)

{

**return** **new** Schet { Summa = schet1.Summa + x, Value = schet1.Value};

}

**public** **static** **bool** operator >(Schet schet1, Schet schet2)

{

**if** (schet1.Summa > schet2.Summa)

**return** true;

**else**

**return** false;

}

**public** **static** **bool** operator <(Schet schet1, Schet schet2)

{

**if** (schet1.Summa < schet2.Summa)

**return** true;

**else**

**return** false;

}

**public** **static** **bool** operator ==(Schet schet1, Schet schet2)

{

**if** (schet1.Summa == schet2.Summa)

**return** true;

**else**

**return** false;

}

**public** **static** **bool** operator !=(Schet schet1, Schet schet2)

{

**if** (schet1.Summa != schet2.Summa)

**return** true;

**else**

**return** false;

}

**public** **static** Schet operator ++(Schet schet1)

{

**return** **new** Schet { Value = schet1.Value, Summa = schet1.Summa +

(schet1.Summa / 100 \* 5) };

}

}

}

## Задание № 2. Разработать класс для хранения методов

Реализация методов представлена в листинге 2.

Листинг 2 – Методы

**using** System;

**using** System.Collections.Generic;

**using** System.Linq;

**using** System.Text;

**using** System.Threading.Tasks;

**namespace** Laba4

{

**class** Methods

{

Schet с1 = **new** Schet();

Schet с2 = **new** Schet();

**public** **void** zap()

{

Console.WriteLine("Данные о первом счете: ");

с1.Input();

с1.Print();

Console.WriteLine();

Console.WriteLine("Данные о втором счете: ");

с2.Input();

с2.Print();

Console.WriteLine();

}

**public** **void** kol()

{

Console.WriteLine();

Console.WriteLine("Слияние двух счетов");

Console.WriteLine("Сумма первого и второго счетов:");

**if** (с1.Value == с2.Value)

{

Schet schet3 = с1 + с2;

schet3.Print();

}

**else**

{

Console.WriteLine("Валюты счетов не совпадают,

операция слияния недоступна");

}

}

**public** **void** snyatie()

{

Console.WriteLine();

Console.Write("Выберите счёт для снятия денег(1,

2): ");

M1: **int** a = 0;

a = Convert.ToInt16(Console.ReadLine());

**if** (a == 1 | a == 2)

{

**if** (a == 1)

{

Console.WriteLine();

Console.WriteLine("Введите сумму денег для

снятия с первого счёта: ");

**double** t =

Convert.ToDouble(Console.ReadLine());

с1 -= t;

с1.Print();

}

**if** (a == 2)

{

Console.WriteLine();

Console.WriteLine("Введите сумму денег для

снятия со второго счёта: ");

**double** t =

Convert.ToDouble(Console.ReadLine());

с2 -= t;

с2.Print();

}

}

**else**

{

Console.WriteLine("Такого счёта нет. Введите

существующий счёт ");

**goto** M1;

}

}

**public** **void** popol()

{

Console.WriteLine();

Console.Write("Выберите счёт для пополнения

деньгами(1, 2): ");

M2: **int** b = 0;

b = Convert.ToInt16(Console.ReadLine());

**if** (b == 1 | b == 2)

{

**if** (b == 1)

{

Console.WriteLine();

Console.WriteLine("Введите сумму для

пополнения первого счёта: ");

**double** x =

Convert.ToDouble(Console.ReadLine());

с1 += x;

с1.Print();

}

**if** (b == 2)

{

Console.WriteLine();

Console.WriteLine("Введите сумму для

пополнения второго счёта: ");

**double** x =

Convert.ToDouble(Console.ReadLine());

с2 += x;

с2.Print();

}

}

**else**

{

Console.WriteLine("Такого счёта нет. Введите

существующий счёт ");

**goto** M2;

}

}

**public** **void** srav()

{

Console.WriteLine();

Console.WriteLine("Сравнение счетов: ");

**if** (с1 > с2)

{

Console.WriteLine();

Console.WriteLine("Первый счёт больше

второго.");

}

**else**

{

**if** (с1 < с2)

Console.WriteLine("Первый счёт меньше

второго.");

**else**

Console.WriteLine("Первый и второй счета

равны.");

}

}

**public** **void** nachis()

{

Console.WriteLine();

Console.Write("Выберите счёт для начисления

процентов(1, 2): ");

M3: **int** c = 0;

c = Convert.ToInt16(Console.ReadLine());

**if** (c == 1 | c == 2)

{

**if** (c == 1)

{

Console.WriteLine("Начисление процентов на

первый счёт(5%): ");

с1++;

с1.Print();

}

**if** (c == 2)

{

Console.WriteLine("Начисление процентов на

второй счёт(5%): ");

с2++;

с2.Print();

}

}

**else**

{

Console.WriteLine("Такого счёта нет. Введите

существующий счёт ");

**goto** M3;

}

}

**public** **void** start()

{

Methods method = **new** Methods();

**for** (**int** z = 0; z < 99999; z++)

{

Console.WriteLine("1. Создать");

Console.WriteLine("2. Слияние счетов");

Console.WriteLine("3. Снятие денег со счета");

Console.WriteLine("4. Пополнение счета");

Console.WriteLine("5. Сравнение счетов");

Console.WriteLine("6. Начисление

процентов(5%)");

Console.WriteLine("7. Выйти.");

Console.Write("Выберите действе: ");

**int** x = Convert.ToInt32(Console.ReadLine());

Console.WriteLine();

**switch** (x)

{

**case** 1: method.zap(); **break**;

**case** 2: method.kol(); **break**;

**case** 3: method.snyatie(); **break**;

**case** 4: method.popol(); **break**;

**case** 5: method.srav(); **break**;

**case** 6: method.nachis(); **break**;

**case** 7: Environment.Exit(0); **break**;

**default**: Console.WriteLine("Введите числа

от 1 до 7"); **break**;

}

}

}

}

}

## Задание № 2. Разработать код программы

Реализация кода программы представлена в листинге 3.

Листинг 3 – Код программы

**using** System;

**using** System.Collections.Generic;

**using** System.Linq;

**using** System.Text;

**using** System.Threading.Tasks;

**namespace** Laba4

{

**class** Program

{

**static** **void** Main(**string**[] args)

{

Methods schet = **new** Methods();

schet.start();

}

}

}

Работа программы предоставлена на рисунках 3.1 – 3.7

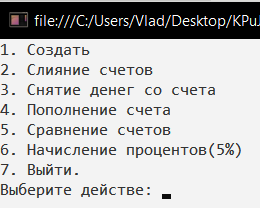


Рисунок 3.1 – Главное меню

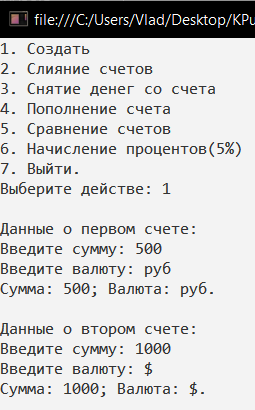


Рисунок 3.2 – Добавление счета

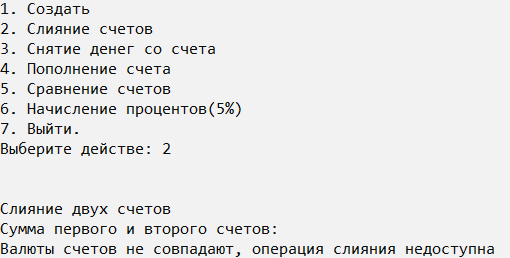


Рисунок 3.3 – Слияние счетов

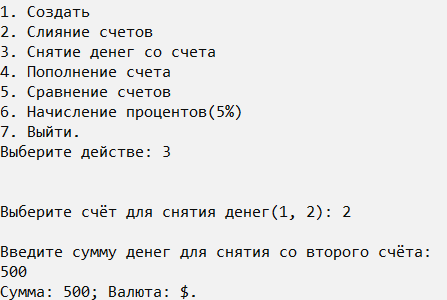


Рисунок 3.4 – Снятие денег со счета

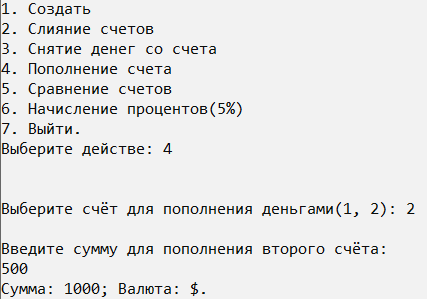


Рисунок 3.5 – Пополнение счета

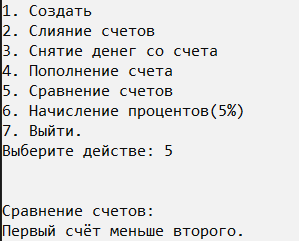


Рисунок 3.6 – Сравнение счетов

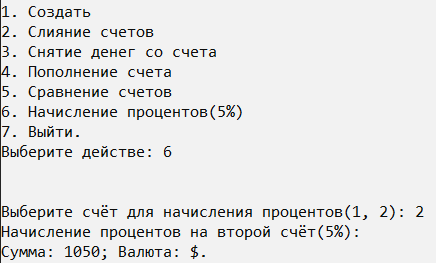


Рисунок 3.7 – Начисление 5%

**Выводы:**

В данной практической работе была выполнена программа содержащая 3 класса: «Schet.cs», «Program.cs», «Methods.cs» которая выполняет различные функции, также листинг который отображен в отчете к практической работе. Я научился разрабатывать программу согласно своему индивидуальному варианту задания.