# Лабораторная работа № 6 – Перегрузка операторов

**1 Цель работы**

Изучить основные принципы перегрузки операторов в C#.

1. **Порядок выполнения работы**
   * прочесть краткие теоретические сведения;
   * выполнить задания раздела;
   * составить отчет по лабораторной работе и защитить его у преподавателя.
2. **Перегрузка операторов**

Встроенные операторы можно перегрузить для более удобного и естественного взаимодействия с пользовательскими типами.

**Перегружены могут быть следующие операторы:**

+ - \* / ++ -- ! ~ % & | ^ == != < >

а также *операторы автоматического и явного приведения* (с помощью ключевых слов implicit и explicit.

*Составные операторы присвоения* (например, +=, /=) **автоматически** переопределяются при переопределении соответствующих не составных операторов (например, +, /).

1. **Операторные методы (Operator Functions)**

Перегрузить оператор можно при помощи объявления операторного метода. Операторный метод должен быть статическим, *и как минимум один из его аргументов должен быть того типа, в котором объявлен операторный метод*. Объявляется операторный метод с помощью ключевого слова operator.

В качестве примера рассмотрим приложение для заказа такси, в котором реализован класс, ответственный за поездку, с тремя полями – «Расстояние», «Длительность поездки» и «Цена».

public class Order

{

public double Distance;

public double Time;

public double Cost;

public Order(double distance, double time, double cost)

{

Distance = distance;

Time = time;

Cost = cost;

}

}

Поскольку наша служба заказа такси недавно появилась на рынке транспортных услуг, часто клиентам предоставляются скидки. Перегрузим оператор «-» для более удобного предоставления скидки:

public class Order

{

public double Distance;

public double Time;

public double Cost;

public Order(double distance, double time, double cost)

{

Distance = distance;

Time = time;

Cost = cost;

}

**public static Order operator - (Order x, int discount)**

**{**

**return new Order(x.Distance, x.Time, x.Cost - x.Cost/100\*discount);**

**}**

}

Теперь в коде есть возможность очень просто предоставить клиенту скидку в 10% на его поездку:

var order = new Order(10, 15, 150);

order -= 10;

1. **Перегрузка операторов равенства и сравнения**

Операторы равенства и сравнения обычно переопределяются для структур, реже для классов. При перегрузке этих операторов необходимо соблюдать несколько правил:

*Парность операторов*: операторы, представляющие логическую пару (== !=), (< >), (<= >=), перегружаются совместно.

Equals и GetHashCode: при перегрузке операторов == и !=, также необходимо переопределить методы Equals и GetHashCode (определены для object), чтобы коллекции и хэштаблицы могли корректно работать с типом.

Для того, чтобы показывать водителю поступающие заказы, ранжированные по расстоянию, было бы неплохо перегрузить операторы сравнения для класса Order:

public class Order

{

public double Distance;

public double Time;

public double Cost;

public Order(double distance, double time, double cost)

{

Distance = distance;

Time = time;

Cost = cost;

}

public static Order operator -(Order x, int discount)

{

return new Order(x.Distance, x.Time, x.Cost - x.Cost/100\*discount);

}

**public static bool operator >(Order x, Order y)**

**{**

**return x.Distance > y.Distance;**

**}**

**public static bool operator <(Order x, Order y)**

**{**

**return x.Distance < y.Distance;**

**}**

}

1. **Автоматическое и явное преобразование**

Автоматическое и явное преобразование могут быть перегружены также как операторы. Перегрузка преобразований типов обычно используется для того, чтоб сделать ее более лаконичной и естественной для связанных типов.

Для удобства расчета заработной платы водителя перегрузим автоматическое преобразование типа Order к цене (double):

…

public static implicit operator double (Order x)

{

return x.Cost;

}

…

Теперь при применении к типу Order операторов, характерных для элементарных типов, будет выполнено автоматическое преобразование типа.

Следующий код выведет суммарную стоимость трех поездок, поскольку оператор «+» не был перегружен, но является характерным для элементарных типов, что повлекло за собой автоматическое преобразование к типу double по описанной выше логике:

var order1 = new Order(10, 15, 150);

var order2 = new Order(9, 12, 130);

var order3 = new Order(22, 43, 408);

Console.WriteLine(order1 + order2 + order3);

Однако стоит обратить внимание на то, что оператор «+» мог быть перегружен. В таком случае, операция бы выполнилась в соответствии с логикой операторного метода.

Для предварительной оценки стоимости и продолжительности поездки на основании дистанции перегрузим явное преобразование:

…

public static explicit operator Order(double distance)

{

return new Order(distance, distance/0.5, 50 + distance \* 10);

}

…

Таким образом, при помощи перегрузки явного преобразования типов мы можем предварительно рассчитать детали поездки, имея значение дистанции и выполнив приведение типа double к типу Order по описанной выше логике:

…

Order order = (Order)10.5;

…

1. **Общее задание для выполнения на паре**

Разработать приложение, реализующее логику службы доставки интернет-магазина, в котором имеется класс «Заказ» с полями «Цена товара», «Цена доставки», «Название товара», «Вес товара».

Перегрузить:

- Операторы сравнения (по весу товара);

- Оператор + (при сложении двух заказов возвращается новый экземпляр класса «Заказ», у которого «Вес товара», «Цена товара» равен сумме соответствующих полей слагаемых товаров, «Название товара» является строкой, в которой перечислены через запятую названия слагаемых товаров, а «Цена доставки» - равна наибольшей из двух слагаемых товаров);

- Автоматическое преобразование типов (к типу double, цена товара + цена доставки).

1. **Индивидуальные задания для выполнения работы**
2. Разработать класс «Химический элемент», содержащий поля «Название», «Формула». Перегрузить для него оператор «+», при сложении двух элементов возвращающий их сочетание в соответствии с законами химии (прописать логику соединения для 5 элементов на выбор).
3. Разработать класс «Оружие» для компьютерной игры, содержащий поля «Название», «Вес», «Урон». Перегрузить:

- Операторы сравнения (по урону);

- Автоматическое преобразование типов (к double, вес).

1. Разработать класс «Ингредиент» для компьютерной игры, содержащий поля «Название», «Эффект», «Цена». Перегрузить:

- Операторы сравнения (по цене);

- Оператор +, при сложении двух ингредиентов возвращающий третий под названием «Зелье», цена равняется сумме слагаемых ингредиентов, помноженных на 3, эффект указывается на основании воображения студента (прописать логику соединения для 5 ингредиентов).

1. Разработать класс «Персонаж» для компьютерной игры, содержащий поля «Имя», «Уровень», «Очки здоровья». Перегрузить:

- Операторы сравнения (по уровню);

- Автоматическое преобразование типов (к double - здоровье).

1. Разработать класс «Элемент». При помощи перегрузки оператора «+» реализовать логику игры «Алхимия» (описание игры и рецепты соединений доступны в сети Интернет).
2. Разработать класс «Предмет». При помощи перегрузки оператора «+» реализовать логику создания предметов для компьютерной игры (объединение двух предметов должно давать третий, прописать не менее 5 соединений).
3. Разработать класс «Карта», содержащий поля «Название», «Масть», «Вес» (int), «Козырь» (bool). Веса распределяются следующим образом: 2 – 10 – в соответствии с номером карты, 11, 12, 13, 14 – валет, дама, король, туз соответственно. При помощи перегрузки операторов сравнения реализовать логику карточной игры (по весу, с учетом козыря).
4. Разработать класс «Карта», содержащий поля «Название», «Масть», «Очки» (int). При помощи перегрузки оператора «+» для сложения карт реализовать логику игры Блэк-Джек (с учетом особенностей туза).
5. Разработать класс «Продукт», содержащий поля «Название», «Вес (г)», «Калорийность на 100г».

Перегрузить:

- Оператор «+» для увеличения веса продукта (с типами аргументов Продукт и double, возвращает новый продукт с весом, увеличенным на значение аргумента double);

- Автоматическое преобразование типов (к double - вес, умноженный на калорийность).

1. Разработать класс «Сериал», содержащий поля «Название», «Количество серий», «Длительность серии». Перегрузить:

- Операторы сравнения (по количеству серий);

- Автоматическое преобразование типов (к double - количество серий, умноженное длительность серии).

1. Разработать класс «Фильм», содержащий поля «Название», «Рейтинг», «Длительность». Перегрузить:

- Операторы сравнения (по рейтингу);

- Автоматическое преобразование типов (к double - длительность).

1. Разработать класс «Книга», содержащий поля «Название», «Год выхода», «Количество страниц». Перегрузить:

- Операторы сравнения (по году выхода);

- Автоматическое преобразование типов (к int – количество страниц).

1. Разработать класс «Акция», содержащий поля «Название компании», «Цена в долларах», «Текущий курс доллара к рублю» (static double). Перегрузить:

- Операторы сравнения (по цене);

- Автоматическое преобразование типов (к double – цена, умноженная на текущий курс).

1. Разработать класс «Курс доллара», содержащий поля «Дата», «Курс доллара к рублю». Перегрузить:

- Операторы сравнения (по дате);

- Автоматическое преобразование типов (к double – курс).

1. Разработать класс «Квантовый скачок», содержащий поля «Энергоемкость скачка» и «Расстояние до Земли». Перегрузить:

- Операторы сравнения (по расстоянию до Земли);

- Оператор «+» (при сложении двух скачков возвращает третий, где расстояние до Земли равно наибольшему расстоянию из двух слагаемых скачков, а энергоемкость – их суммарной энергоемкости).

1. Разработать класс «Временной сдвиг», содержащий поля «Исходный год», «Целевой год» и «Текущий год» (static). Перегрузить:

- Операторы сравнения (по абсолютной разнице между исходным и целевым годом);

- Оператор «+» (при сложении двух сдвигов возвращается третий, где целевой год равен наиболее удаленному целевому году от текущего среди слагаемых сдвигов, а исходный – наиболее близкому к текущему исходному году из двух слагаемых сдвигов).

1. Разработать класс «Силовая установка звездолёта», содержащий поля «Тип» (химическая, плазменная, фотонная и т.п.), «Тяга (кг)». Перегрузить:

- Операторы сравнения (по величине тяги);

- Оператор «+» (при сложении двух силовых установок возвращается третья, где величина тяги равняется сумме тяг слагаемых силовых установок, а тип равняется «Гибридная», если слагаемые силовые установки различаются по типу, либо равен типу слагаемых силовых установок, если их типы равны между собой).

1. Разработать класс «Видеоролик», содержащий поля «Название», «Количество просмотров», «Длительность». Перегрузить:

- Операторы сравнения (по количеству просмотров);

- Автоматическое преобразование типов (к double - длительность).

1. Разработать класс «Пост в социальной сети», содержащий поля «Id», «Количество лайков», «Количество дизлайков». Перегрузить:

- Операторы сравнения (по количеству лайков);

- Оператор «+» (с типами аргументов «Пост в социальной сети» и int, возвращает новый экземпляр класса «Пост в социальной сети», где количество лайков увеличено на значение аргумента int);

- Оператор «-» (с типами аргументов «Пост в социальной сети» и int, возвращает новый экземпляр класса «Пост в социальной сети», где количество дизлайков увеличено на значение аргумента int).

1. Разработать класс «Композиция», содержащий поля «Название», «Группа», «Год», «Длительность». Перегрузить:

- Операторы сравнения (по году);

- Автоматическое преобразование типов (к double - длительность).