

```

delegate int PlusOrMinus(int p1, int p2);
class Program
{
    static int Plus(int p1, int p2)
    {
        return p1 + p2;
    }
    static int Minus(int p1, int p2)
    {
        return p1 - p2;
    }
}

```

Особенность языка C# состоит в том, что делегатный тип может быть использован в качестве параметра метода.

Пример объявления метода с делегатным параметром:

```

static void PlusOrMinusMethod(
    string str,
    int i1,
    int i2,
    PlusOrMinus PlusOrMinusParam)
{
    int Result = PlusOrMinusParam(i1, i2);
    Console.WriteLine(str + Result.ToString());
}

```

последний параметр – это параметр делегатного типа. В теле метода имя этого параметра используется как функция, производится ее вызов:

```
int Result = PlusOrMinusParam(i1, i2);
```

Однако метода PlusOrMinusParam в программе нет. Реально будет вызван метод, переданный в качестве параметра в функцию PlusOrMinusMethod.

Таким образом, параметр делегатного типа вызывается как еще не известный метод, который в будущем будет передан в качестве параметра. Сигнатура вызова PlusOrMinusParam соответствует сигнатуре делегата: передаются два входных параметра целого типа, возвращается значение целого типа.

Пример вызова метода с делегатным параметром:

```

int i1 = 3;
int i2 = 2;
PlusOrMinusMethod("Плюс: ", i1, i2, Plus);
PlusOrMinusMethod("Минус: ", i1, i2, Minus);
Результат вывода в консоль:

```

Плюс: 5

Минус: 1

В данном примере в качестве последнего параметра
PlusOrMinusMethod передаются имена функций Plus и Minus