

클래스 기본 및 메서드

HyoJoon Han
동국대학교
han6343@dongguk.edu



추상화

메서드 기본 형태

매개변수와 반환

오버로딩

- 클래스 기반의 객체 지향 프로그래밍 언어의 특징
 - 추상화, 캡슐화, 상속, 다형성
 - 추상화 : 프로그램에 사용되는 핵심적인 부분을 추출하는 것

코드 5-17 학생 추상화

/5장/InstanceVariables

```
class Student
{
    public string id;
    public string name;
    public int grade;
    public string major;
    public DateTime birthday;

    /* 계속해서 생각해보세요. */
}
```

- 메서드의 기본 형태

```
[접근제한자] [반환형] [메서드 이름]([매개변수])  
{  
    [메서드 코드]  
}
```

이후에 자세히 알아볼 것이므로 일단 public을 입력한다고 생각합시다.

그림 6-1 메서드의 기본 형태

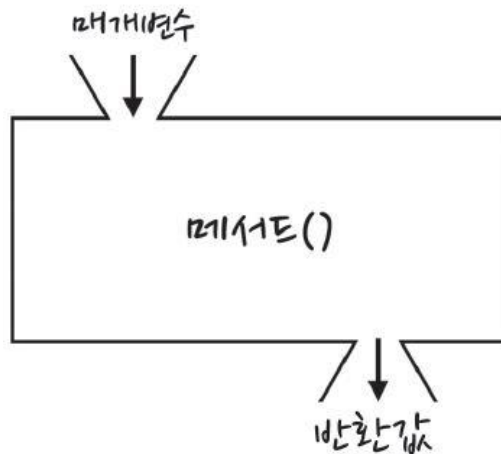


그림 6-2 함수 상자와 메서드

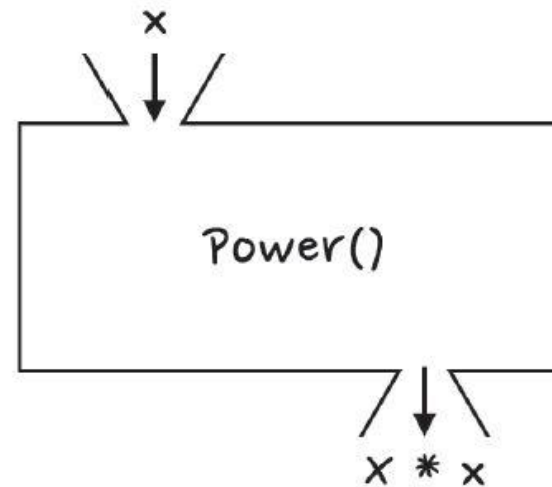
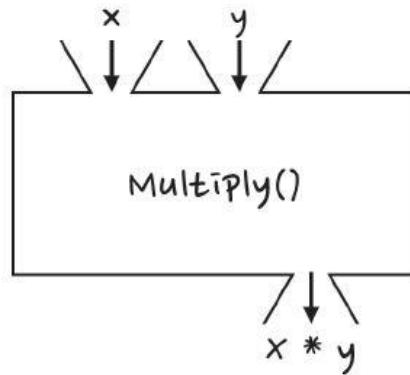


그림 6-3 Power() 메서드

- 메서드는 여러 개의 매개변수를 가질 수 있다

코드 6-2 두 개의 매개변수를 갖는 메서드

/6장/Methods



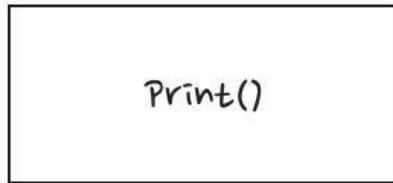
```
class Program
{
    class Test
    {
        public int Multi(int x, int y)
        {
            return x * y;
        }
    }

    static void Main(string[] args)
    {
        Test test = new Test();
        Console.WriteLine(test.Multi(52, 273));
        Console.WriteLine(test.Multi(103, 32));
    }
}
```

- 아무것도 반환하지 않는 메서드

코드 6-3 아무것도 반환하지 않는 메서드

/6장/Methods



```
class Program
{
    class Test
    {
        public void Print()
        {
            Console.WriteLine("Print() 메서드가 호출
되었습니다.");
        }
    }

    static void Main(string[] args)
    {
        Test test = new Test();
        test.Print();
        test.Print();
        test.Print();
    }
}
```

- 반환 메서드 형태

```
public 자료형 메서드(자료형 매개변수, 자료형 매개변수)
{
    자료형 output = 초기값;

    // output에 값을 계산

    return output;
}
```

- 오버로딩(Overloading) : 이름은 같고, 매개변수는 다른 메서드를 만드는 것

Math.Abs()

▲ 1/7 ▼ decimal Math.Abs(decimal value)

System.Decimal 숫자의 절대 값을 반환합니다.

value: System.Decimal.MinValue보다 크거나 같지만 System.Decimal.MaxValue보다 작거나 같은 숫자입니다.

그림 6-5 Math.Abs() 메서드의 형태(1)

Math.Abs()

▲ 3/7 ▼ float Math.Abs(float value)

단정밀도 부동 소수점 수의 절대 값을 반환합니다.

value: System.Single.MinValue보다 크거나 같지만 System.Single.MaxValue보다 작거나 같은 숫자입니다.

그림 6-6 Math.Abs() 메서드의 형태(2)

- 오버로딩 불가능한 예제

코드 6-11 오버로딩 불가

/6장/overloads

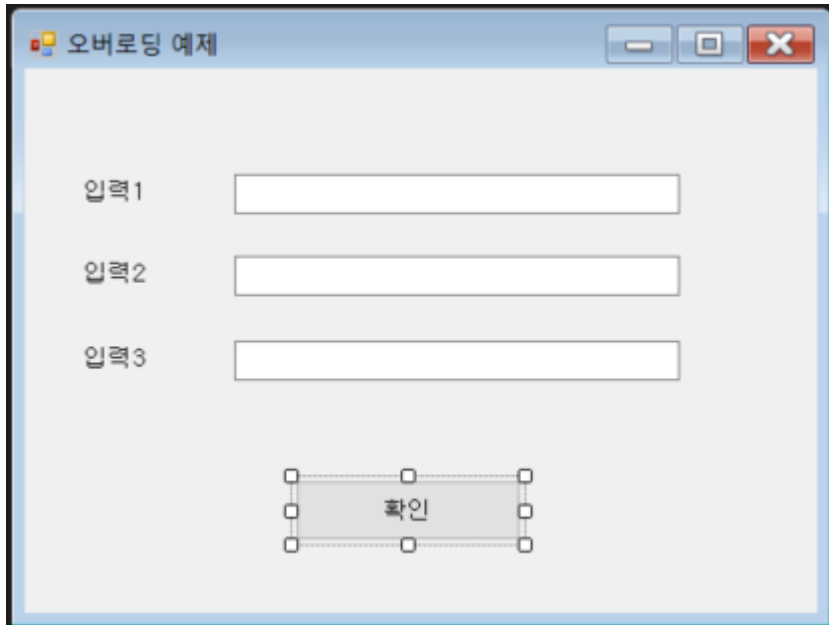
```
class TestWorld
{
    public int Test(int input) { }
    public double Test(int input) { }
    public double Test(int input) { }
}
```

```
class TestWorld
{
    public int Test(int input) { }
    public double Test(int input) { }
    public double Test(int input) { }
}
```

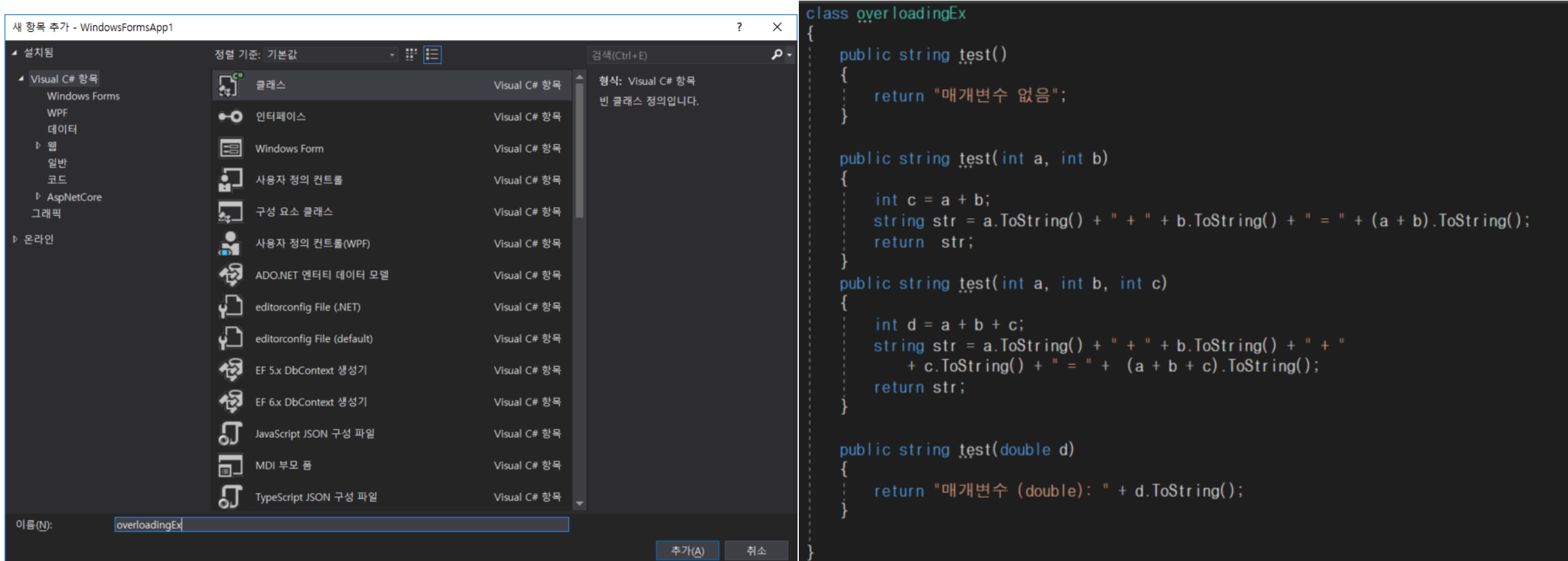
'ClassBasic.Program.TestWorld.Test(int)': 코드 경로 중 일부만 값을 반환합니다.

- 실습 오버로딩 예제

1) 윈도우 폼 디자인



2) 클래스 생성 및 코드 입력



The screenshot shows the Visual Studio IDE. On the left, the 'Add New Item' dialog is open for 'WindowsFormsApp1'. The 'Visual C# 항목' (Visual C# Items) category is selected. The '클래스' (Class) item is highlighted in the list. The '이름(N):' (Name) field at the bottom contains 'overloadingEx'. On the right, the code editor shows the following C# code for the 'overloadingEx' class:

```
class overloadingEx
{
    public string test()
    {
        return "매개변수 없음";
    }

    public string test(int a, int b)
    {
        int c = a + b;
        string str = a.ToString() + " + " + b.ToString() + " = " + (a + b).ToString();
        return str;
    }

    public string test(int a, int b, int c)
    {
        int d = a + b + c;
        string str = a.ToString() + " + " + b.ToString() + " + " +
            + c.ToString() + " = " + (a + b + c).ToString();
        return str;
    }

    public string test(double d)
    {
        return "매개변수 (double): " + d.ToString();
    }
}
```

3) 코드 입력

```
private void button1_Click(object sender, EventArgs e)
{
    string str = "";
    overloadingEx over = new overloadingEx();

    if(textBox1.Text != "")
    {
        if(textBox2.Text != "")
        {
            if(textBox3.Text != "")
            {
                int num1 = int.Parse(textBox1.Text);
                int num2 = int.Parse(textBox2.Text);
                int num3 = int.Parse(textBox3.Text);

                str = over.test(num1, num2, num3);
            }
        }
    }
}
```

```
    else
    {
        int num1 = int.Parse(textBox1.Text);
        int num2 = int.Parse(textBox2.Text);

        str = over.test(num1, num2);
    }
    else
    {
        double num = double.Parse(textBox1.Text);
        str = over.test(num);
    }
    else
    {
        str = over.test();
    }

    MessageBox.Show(str);
}
```

4) 결과 확인

