

Mục tiêu

- Nắm được khái niệm delegate trong C#.
- Nắm được khái niệm event trong C#.

Nội dung

Delegate trong C#

Bài toán

Xây dựng chương trình tính toán với 2 số.

Hướng dẫn

Xây dựng class **MathOperations** với các phương thức hỗ trợ tính toán.

```
class MathOperations
{
    public static decimal Cong(decimal v1, decimal v2)
    {
        return v1 + v2;
    }

    public static decimal Tru(decimal v1, decimal v2)
    {
        return v1 - v2;
    }

    public static decimal Nhan(decimal v1, decimal v2)
    {
        return v1 * v2;
    }

    public static decimal Chia(decimal v1, decimal v2)
    {
        return v1 / v2;
    }
}
```

Cài đặt trong Program

```
class Program
{
    // khai báo delegate phù hợp phương thức tính phép toán
    delegate decimal TinhToan(decimal v1, decimal v2);
    static void Main(string[] args)
    {
        Random rd = new Random();
        decimal v1, v2;
        char c = 'a', pt;
        while (c != 'q')
        {
            TinhToan tt = LuaChonPhepToan(out pt);
            v1 = rd.Next(1000);
            v2 = rd.Next(1000);
            Console.WriteLine("\n{0} {1} {2} = {3}", v1, pt, v2, tt(v1, v2));
        }
    }
}
```

```

        Console.WriteLine("Tiep tuc?");
        c = Console.ReadKey().KeyChar;
        Console.WriteLine();
    }
}

// phương thức lựa chọn phép toán
static TinhToan LuaChonPhepToan(out char phepToan)
{
    // delegate giữ hàm tính toán
    TinhToan tt = null;
    Console.WriteLine("Lua chon phep toan: ");
    phepToan = Console.ReadKey().KeyChar;
    switch (phepToan)
    {
        case '+':
            tt = new TinhToan(MathOperations.Cong);
            break;
        case '-':
            tt = new TinhToan(MathOperations.Tru);
            break;
        case '*':
            tt = new TinhToan(MathOperations.Nhan);
            break;
        case '/':
            tt = new TinhToan(MathOperations.Chia);
            break;
    }
    return tt;
}
}

```

Multicasting với Delegate trong C#

Bài toán

Viết chương trình nhập vào và xuất kết quả các phép toán.

Hướng dẫn

Xây dựng class **XL**TinhToan

```

class XLTinhToan
{
    public static void PhepCong(decimal v1, decimal v2)
    {
        Console.WriteLine("{0} + {1} = {2}", v1, v2, v1 + v2);
    }
    public static void PhepTru(decimal v1, decimal v2)
    {
        Console.WriteLine("{0} - {1} = {2}", v1, v2, v1 - v2);
    }
    public static void PhepNhan(decimal v1, decimal v2)
    {
        Console.WriteLine("{0} x {1} = {2}", v1, v2, v1 * v2);
    }
    public static void PhepChia(decimal v1, decimal v2)
    {
        Console.WriteLine("{0} : {1} = {2}", v1, v2, v1 / v2);
    }
}

```

Cài đặt trong Program

```
class Program
{
    // khai báo delegate phù hợp multicast
    delegate void KQTinhToan(decimal v1, decimal v2);
    static void Main(string[] args)
    {
        // tạo delegate với multicasting
        // chú ý đây là cú pháp ngắn gọn
        KQTinhToan tt = XLTinhToan.PhepCong;
        tt += XLTinhToan.PhepTru;
        tt += XLTinhToan.PhepNhan;
        tt += XLTinhToan.PhepChia;
        decimal v1, v2;
        char c = 'a';
        while (c != 'q')
        {
            Console.Write("Nhap vao 2 so: ");
            v1 = NhapSo();
            v2 = NhapSo();
            // chỉ invoke delegate 1 lần
            tt(v1, v2);
            Console.Write("Tiep tuc?");
            c = Console.ReadKey().KeyChar;
            Console.WriteLine();
        }
    }

    static decimal NhapSo()
    {
        decimal n = 0;
        string strLine = Console.ReadLine();
        if (!decimal.TryParse(strLine, out n))
        {
            n = -1;
        }
        return n;
    }
}
```

Ứng dụng delegate và anonymous

Bài toán

Xây dựng chương trình cho phép sắp xếp danh sách Sinh viên (gồm MSSV, HoTen, QueQuan) theo từng tiêu chí riêng.

Hướng dẫn

Tạo class **SinhVien**

```
class SinhVien
{
    int _MSSV;
    string _hoTen, _queQuan;
    public int MSSV
    {
        get { return _MSSV; }
        set { _MSSV = value; }
    }
}
```

```

public string QueQuan
{
    get { return _queQuan; }
    set { _queQuan = value; }
}
public string HoTen
{
    get { return _hoTen; }
    set { _hoTen = value; }
}
public override string ToString()
{
    return string.Format("<{0}>-{1}-[{2}]", _MSSV, _hoTen, _queQuan);
}
}

```

Cài đặt trong Program

```

class Program
{
    static void Main(string[] args)
    {
        Random rd = new Random();
        // phát sinh số lượng sv
        int n = rd.Next(5, 10);
        SinhVien[] dsSV = new SinhVien[n];
        for (int i = 0; i < n; i++)
        {
            // cách khởi tạo nhanh
            dsSV[i] = new SinhVien()
            {
                MSSV = 1364000 + rd.Next(999),
                HoTen = string.Format("{0}HT", (char)rd.Next(0x41, 0x5A)),
                QueQuan = string.Format("{0}QQ", (char)rd.Next(0x61, 0x7A))
            };
        }
        char c = 'a';
        while (c != 'q')
        {
            Console.WriteLine("Sap xep theo tieu chi (1,2,3): ");
            switch (Console.ReadKey().KeyChar)
            {
                case '1':
                    // sắp xếp theo MSSV
                    Array.Sort(dsSV, delegate(SinhVien sv1, SinhVien sv2)
                    {
                        return sv1.MSSV.CompareTo(sv2.MSSV);
                    });
                    break;
                case '2':
                    // sắp xếp theo HoTen
                    Array.Sort(dsSV, delegate(SinhVien sv1, SinhVien sv2)
                    {
                        return sv1.HoTen.CompareTo(sv2.HoTen);
                    });
                    break;
                case '3':
                    // sắp xếp theo QueQuan
                    Array.Sort(dsSV, delegate(SinhVien sv1, SinhVien sv2)
                    {
                        return sv1.QueQuan.CompareTo(sv2.QueQuan);
                    });
                    break;
            }
            c = Console.ReadKey().KeyChar;
        }
    }
}

```

```

        break;
    }
    Console.WriteLine();
    foreach (var item in dsSV)
    {
        Console.WriteLine(item);
    }
    Console.Write("Tiep tuc?");
    c = Console.ReadKey().KeyChar;
    Console.WriteLine();
}
}
}

```

Event trong C#

Bài toán

Xây dựng chương trình tiếp thị các mặt hàng phù hợp với nhu cầu khi một đối tượng có mong muốn mua sắm trong khoản ngân sách.

Hướng dẫn

Xây dựng class **KhachHang** (sẽ có nhu cầu mua sắm trong khoản ngân sách cho phép: bình dân, trung cấp và cao cấp).

```

class KhachHang
{
    // khai báo delegate
    public delegate void HangMucHandler();
    // khai báo event
    public event HangMucHandler DatMucBinhDan;
    public event HangMucHandler DatMucTrungCap;
    public event HangMucHandler DatMucCaoCap;

    static long mucTrungCap = 10000000;
    static long mucCaoCap = 100000000;
    long _taiKhoan = 0;
    // chú ý phương thức set
    public long TaiKhoan
    {
        get { return _taiKhoan; }
        set
        {
            _taiKhoan = value;
            // kiểm tra hạng mục và đưa ra mong muốn mua sắm
            if (_taiKhoan < mucTrungCap)
            {
                // kích hoạt event
                if (DatMucBinhDan != null)
                {
                    DatMucBinhDan();
                }
            }
            else
            {
                if (_taiKhoan < mucCaoCap)
                {
                    if (DatMucTrungCap != null)
                    {
                        DatMucTrungCap();
                    }
                }
            }
        }
    }
}

```


Bài tập

Bài 1

Hoàn thành các bài ví dụ ở trên.

Bài 2

Tự đề xuất và viết chương trình minh họa thêm một ví dụ về event.