

BÀI THỰC HÀNH 1: Ứng dụng Calculator với ASP.NET MVC

Phần 1: Tạo ứng dụng



Nhập các số hạng, click các button để tính toán:



Bước 1: Tạo MVC project “Calculator”

Bước 2: Trong Model, tạo class Expression

```
public class Expression
{
    public double Number1 { get; set; }
    public double Number2 { get; set; }
    public double Result { get; set; }
    public Expression()
    {
        this.Result = 0;
    }
}
```

Bước 3: Thêm một Action Index với phương thức HttpPost trong HomeController

```
[HttpPost]
public ActionResult Index(Expression model, string command)
{
    switch (command)
    {
        case "+": model.Result = model.Number1 + model.Number2; break;
        case "-": model.Result = model.Number1 - model.Number2; break;
        case "x": model.Result = model.Number1 * model.Number2; break;
        case "/":
            if (model.Number2 != 0)
                model.Result = model.Number1 / model.Number2;
            break;
    }
}
```

```

    }
    return View(model);
}

```

Bước 4: Hiệu chỉnh trang Home\Index.cshtml

```

@model Calculator.Models.Expression
@{
    ViewBag.Title = "Các phép tính đơn giản";
}
<h2>@ViewBag.Title</h2>

@using (Html.BeginForm())
{
    <table class="table-borderless">
        <tr>
            <th>@Html.Label("Số thứ 1")</th>
            <td>
                @Html.EditorFor(model => model.Number1)
            </td>
        </tr>
        <tr>
            <th>@Html.Label("Số thứ 2")</th>
            <td>
                @Html.EditorFor(model => model.Number2)
            </td>
        </tr>
        <tr>
            <th>
            </th>
            <td>
                <input class="btn btn-primary" type="submit" value="+" name="command" />
                <input class="btn btn-primary" type="submit" value="-" name="command" />
                <input class="btn btn-primary" type="submit" value="x" name="command" />
                <input class="btn btn-primary" type="submit" value="/" name="command" />
            </td>
        </tr>
        <tr>
            <th>
                <h3 style="color: red"> @Html.Label("Kết quả: ")</h3>
            </th>
            <td><h3 style="color: red"> @Html.DisplayFor(model => model.Result)</h3></td>
        </tr>
    </table>
}

```

Thực thi ứng dụng, lần lượt nhập các số, thực hiện các phép tính và kiểm tra kết quả

Bước 5: Kiểm chứng dữ liệu, bắt buộc nhập

- Hiệu chỉnh class Expression

```

public class Expression
{
    [Required(ErrorMessage = "Phải nhập một số")]
    public double Number1 { get; set; }
    [Required(ErrorMessage = "Phải nhập một số")]

```

```
public double Number2 { get; set; }
public double Result { get; set; }
public Expression()
{
    this.Result = 0;
}
}
```

- **Hiệu chỉnh trong Controller**

```
[HttpPost]
public ActionResult Index(Expression model, string command)
{
    if (ModelState.IsValid)
    {
        switch (command)
        {
            case "+": model.Result = model.Number1 + model.Number2; break;
            case "-": model.Result = model.Number1 - model.Number2; break;
            case "x": model.Result = model.Number1 * model.Number2; break;
            case "/":
                if (model.Number2 != 0)
                    model.Result = model.Number1 / model.Number2;
                break;
        }
        return View(model);
    }
    return View();
}
```

- **Hiệu chỉnh trong View**

```
<tr>
  <th>@Html.Label("Số thứ 1")</th>
  <td>
    @Html.EditorFor(model => model.Number1)
    <span class="text-danger">@Html.ValidationMessageFor(model => model.Number1)</span>
  </td>
</tr>
<tr>
  <th>@Html.Label("Số thứ 2")</th>
  <td>
    @Html.EditorFor(model => model.Number2)
    <span class="text-danger">@Html.ValidationMessageFor(model => model.Number2)</span>
  </td>
</tr>
```

Phần 2: Phát triển ứng dụng với giao diện như sau:

Các phép tính đơn giản

+ =

Cho phép chọn phép tính:

Bước 1: Hiệu chỉnh class Expression

```
public class Expression
{
    [Required(ErrorMessage = "Phải nhập một số")]
    public double Number1 { get; set; }
    [Required(ErrorMessage = "Phải nhập một số")]
    public double Number2 { get; set; }
    public double Result { get; set; }
    public string Operator { get; set; }
    public Expression()
    {
        this.Result = 0;
    }
}
```

Bước 2: Tạo TotalController

```
public class TotalController : Controller
{
    // GET: Total
    public ActionResult Index(Expression exp)
    {
        return View(exp);
    }
    [HttpPost]
    public ActionResult Calculate(Expression exp)
    {
        exp.Result = CalculateResult(exp);
        return RedirectToAction("Index", exp);
    }

    private double CalculateResult(Expression exp)
    {
        double result = 0;
        switch (exp.Operator)
        {
            case "+":
                result = exp.Number1 + exp.Number2;
                break;
            case "-":
                result = exp.Number1 - exp.Number2;
                break;
            case "x":
                result = exp.Number1 * exp.Number2;
                break;
            case "/":
                result = exp.Number1 / exp.Number2;
                break;
        }
    }
}
```

```

        if (exp.Number2 != 0)
            result = exp.Number1 / exp.Number2;
            break;
    }
    return result;
}
}

```

Bước 3: Tạo View tương ứng

[illegible]

Thực thi ứng dụng với liên kết: localhost:port /Total/Index