Упражнение 1. Использование перечислений

В этом упражнении вы определите перечисления, которые представляют различные материалы (нержавеющая сталь, алюминий, железобетон и титан) и поперечное сечение балок (двутавровая балка, Коробка, Z-образная и C-образная). Кроме того, будет определен еще одно перечисление, которое называется TestResult, представляюще собой результаты stress теста.

Основными задачами данного упражнения являются:

1. Откройте решение перечисления.

2. Добавьте перечисления в пространство имен StressTest.

3. Извлеките значения перечисления.

4. Отображение результатов выбора.

5. Протестируйте решение.

Задача 1. Откройте решение перечислений

1. Откройте решение перечислений в папке Ex1\Starter.

Задача 2. Добавление перечислений в пространство имен StressTest

1. Просмотрите список задач.

2. Найдите задачу перечисления Todo-Implement Material, CrossSection and TestResult и щелкните эту задачу. Эта задача находится в StressTestType.cs.

3. В пространстве имен StressTest определите новое перечисление с именем Material. Перечисления должны иметь следующие значения:

|  |  |
| --- | --- |
| a. | **StainlessSteel** |
| b. | **Aluminum** |
| c. | **ReinforcedConcrete** |
| d. | **Composite** |
| e. | **Titanium** |

4. Под перечислением Material определите новое перечисление с именем CrossSection. Перечисление должно иметь следующие значения:

|  |  |
| --- | --- |
| a. | **IBeam** |
| b. | **Box** |
| c. | **ZShaped** |
| d. | **CShaped** |

5. Под перечислением CrossSection определите новое перечисление с именем TestResult. Перечисление имеет следующие значения:

|  |  |
| --- | --- |
| a. | **Pass** |
| b. | **Fail** |

6. Постройте решение и исправьте все ошибки.

Задача 3: Восстановление значений перечисления

1. В проекте TestHarness отобразите окно MainWindow.xaml .

Цель проекта TestHarness - предоставить возможность отображения значений каждого перечисления. При запуске приложения, три списка будут заполнены значениями, определенными для каждого перечисления. Пользователь может выбрать элемент из каждого списка, и приложение создаст строку из соответствующих перечислений.

2. В списке задач найдите **TODO - Retrieve user selections from the UI**, а затем дважды щелкните мышью на ней. Эта задача находится в окне **MainWindow**.**xaml**.**cs** .

3. Удалите комментарий и добавьте код в метод selectionChanged для выполнения следующих задач:

a. Создайте объект материала с именем selectedMaterial и инициализируйте его значением свойства SelectedItem из списка **materials.**

b. Создайте объект CrossSection с именем selectedCrossSection и инициализируйте его значением свойства SelectedItem из списка crosssections.

c. Создайте объект TestResult с именем selectedTestResult и инициализируйте его значением SelectedItemproperty из списка testresults.

Подсказка: свойство SelectedItem элемента управления ListBox имеет Тип объекта. Необходимо привести это свойство к соответствующему типу при назначении его переменной перечисления.

Задача 4. Отображение результатов выбора

1. В методе selectionChanged после кода, добавленного в предыдущей задаче, добавьте инструкцию создания нового объекта StringBuilder с именем selectionStringBuilder.

2. Добавьте оператор switch для оценки переменной selectedMaterial. В инструкции switch добавьте предложения case для каждого потенциального значения перечисления Material. В каждом операторе Case, добавьте код для добавления текста "Материал: <selectedMaterial>," к объекту selectionStringBuilder. Замените текст "<selectedMaterial> " в этой строке соответствующим значением переменной selectedMaterial, которая находится в следующей таблице.

| **Material enumeration value** | **<*selectedMaterial*> string** |
| --- | --- |
| **Material.StainlessSteel** | Stainless Steel |
| **Material.Aluminum** | Aluminum |
| **Material.ReinforcedConcrete** | Reinforced Concrete |
| **Material.Composite** | Composite |
| **Material.Titanium** | Titanium |

3. Добавьте другой оператор switch для оценки переменной selectedCrossSection. В этой инструкции switch добавьте операторы case для каждого потенциального значения перечисления CrossSection. В каждом операторе case добавьте код, чтобы добавить текст "Cross-section: <selectedCrossSection>" к объекту selectionStringBuilder. Замените текст "<selectedCrossSection> " в этой строке соответствующим значением переменной selectedCrossSection, которое показано в следующей таблице.

| **Material enumeration value** | **<*selectedCrossSection*> string** |
| --- | --- |
| **CrossSection**.**IBeam** | I-Beam |
| **CrossSection**.**Box** | Box |
| **CrossSection**.**ZShaped** | Z-Shaped |
| **CrossSection**.**CShaped** | C-Shaped |

4. Добавьте заключительный оператор switch для оценки члена selectedTestResult. В инструкции switch добавьте предложения case для каждого потенциального значения перечисления TestResult. В каждой инструкции case, добавить код для присоединения текста "результат: <selectedTestResult>." к объекту selectionStringBuilder. Замените текст "<selectedTestResult> " в этой строке соответствующим значением переменной selectedTestResult, которая находится в следующей таблице:

| **Material enumeration value** | **<*selectedTestResult*> string** |
| --- | --- |
| **TestResult.Pass** | Pass |
| **TestResult.Fail** | Fail |

5. В конце метода selectionChanged добавьте код для отображения строки, созданной с помощью объекта selectionStringBuilder в свойстве Content метки testDetails.

Задание 5: проверка решения

1. Постройте приложение и исправьте все ошибки.

2. Запустите приложение.

3. В окне mainwindow, в списке **Material**, нажмите титан, в списке **CrossSection**, щелкните **Box**, а затем в списке результатов, нажмите кнопку не получится.

В нижней части окна убедитесь, что метка обновляется вместе с выбранными параметрами.

4. Поэкспериментируйте, выбрав дополнительные значения из всех трех списков, и убедитесь, что при каждом изменении labelupdates отражает изменения.

5. Закройте приложение и вернитесь в Visual Studio.