Transformation Priority Premise

Reto Lehre

Content

- What is TPP
- Transformation List
- Examples
- Review

What is TPP

- It gives you a guidance on how to apply small transformations to the code
- TPP avoids that you take to big steps in TDD
- Transformations on the top of the list should be preferred to those that are lower
- red -> green -> refactor

Transformation List

#	TRANSFORMATION	STARTING CODE	FINAL CODE
1	{} => **nil**		`return nil`
2	nil => **constant**	`return nil`	`return "1"`
3	constant => **constant+**	`return "1"`	`return "1" + "2"`
4	constant => **scalar**	`return "1" + "2"`	`return argument`
5	statement => **statements**	`return argument`	`return arguments`
6	unconditional => **conditional**	`return arguments`	`if(condition)return arguments`
7	scalar => **array**	`dog`	`[dog, cat]`
8	array => **container**	`[dog, cat]`	`{dog = "DOG", cat = "CAT"}`
9	statement => **tail recursion**	`a + b`	`a + recursion`
10	conditional => **loop**	`if(condition)`	`while(condition)`
11	tail recursion => **full recursion**	`a + recursion`	`recursion`
12	expression => **function**	`today - birthday`	`CalculateAge()`
13	variable => **mutation**	`day`	`var day = 10; day = 11;`
14	**switch case**		

The List is ordered by ascending complexity from top to down.
Transformations on top of the List should be preferred.

Examples

```
public class RomanConverter
{
    public string convert(int number)
    {
        return null;
    }
}
```

nil -> constant


```
public class RomanConverter
{
    public string Convert(int number)
    {
        return "I";
    }
}
```

```
public class RomanConverter
{
    public String convert(int number)
    {
        return "I";
    }
}
```

contant -> constant+


```
public class RomanConverter
{
    public String convert(int number)
    {
        return "I" + " V";
    }
}
```

```
public class RomanConverter
{
    public String convert(int number)
    {
        return "I";
    }
}
```

unconditional -> conditional


```
public class RomanConverter
{
    public string convert(int number)
    {
        String result = "I";

        if (number >= 1)
        {
            result += "I";
        }
        return result;
    }
}
```

Examples

```
public class RomanConverter
{
    public String convert(int number)
    {
        String result = "I";
        if (number > 1)
        {
            result += "I";
        }
        if (number > 2)
        {
            result += "I";
        }
        return result;
    }
}
```

scalar -> array

Rule 7

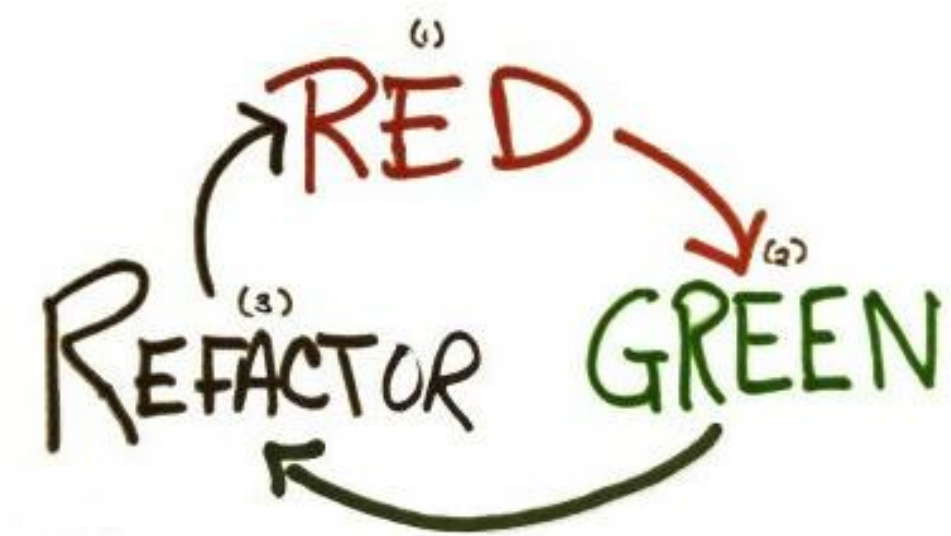
```
public class RomanConverter
{
    private String[] result = { "I", "II", "III" };

    public String Convert(int number)
    {
        return result[number - 1];
    }
}
```

Review

- In the beginning I didn’t realise what TPP is good for
- Good Kata RomanNumbers
- Always have a look to the transformation list
- Code turned better after using TPP

TPP



?