Keeping the Focus on Domain Logic with Sequences

Zoran Horvat
OWNER AT CODING HELMET CONSULTANCY
@zoranh75 codinghelmet.com

```
foreach (IPainter painter in painters)
{
  if (painter.IsAvailable)
  {
    double price = painter.EstimateCompensation(sqMeters);
    if (cheapest == null || price < bestPrice)
    {
      cheapest = painter;
    }
  }
}
```

Code Quality Questionnaire

1. **Can you tell what the code does? – Today, or in a month?**
2. **Can you prove it is correct?**
3. **If there is a bug – can you tell if it is here or somewhere else?**

```
foreach (IPainter painter in painters)
{
  if (painter.IsAvailable)
  {
    double price = painter.EstimateCompensation(sqMeters);
    if (cheapest == null || price < bestPrice)
    {
      cheapest = painter;
    }
  }
}
```

Conclusion

This is truly a bad piece of code.

1. **It is hard to understand.**
2. **There is a gap between requirements and implementation.**

```
private static IPainter FindCheapestPainter(
    double sqMeters, IEnumerable<IPainter> painters)
{
  return
    painters
      .ThoseAvailable()
      .WithMinimum(painter.EstimateCompensation(sqMeters));
}
```

Mind Experiment with Desired Implementation

Sequence of steps when there is a bug reported:

1. **Is the sequence of painters empty?**
2. **Are all painters unavailable?**
3. **If that has nothing to do with the bug, then the bug is elsewhere.**

The Problem of Picking the Best Fit

Given a sequence of N elements,
st fitting one.

Bad Idea: Sorting

Better Idea: Picking

A Word About Coding Culture

correctly or incorrectly.

unexpected ways.

surprise fellow coders.

The Principle of Least Surprise (a.k.a. The Principle of Least Astonishment)

“If a necessary feature has a high astonishment factor, it may be necessary to redesign the feature.”

Mike Cowlishaw

Summary

Use of `for/foreach/while` is disputable

- Too easy to make a bug inside a loop

Structure of the loop

- Loop itself is infrastructure
- Operations inside a loop are logic

Removing loops

- Let the sequence loop through itself
- Only keep control of the operations executed on each of the elements

Use LINQ to Objects to replace loops

Comparison of Methods

With explicit foreach loop

```
private static IPainter FindCheapestPainter(
    double sqMeters, IEnumerable<IPainter> painters)
{
    double bestPrice = 0;
    IPainter cheapest = null;

    foreach (IPainter painter in painters)
    {
        if (painter.IsAvailable)
        {
            double price =
                painter.EstimateCompensation(sqMeters);

            if (cheapest == null || price < bestPrice)
            {
                cheapest = painter;
            }
        }
    }

    return cheapest;
}
```

With LINQ extension methods

```
private static IPainter FindCheapestPainter(
    double sqMeters, IEnumerable<IPainter> painters)
{
    return
        painters
            .ThoseAvailable()
            .WithMinimum(painter =>
                painter.EstimateCompensation(sqMeters));
}
```

Next module –

Encapsulating Data Structure Knowledge inside Collections