Express Yourself!

Alastair Smith ~ I T.A.K.E. Unconference 2016
@alastairs ~ alastair@alastairsmith.me.uk

“You’ll never know less than you know right now”

– *Kent Beck*

https://www.flickr.com/photos/quinnanya/5906875015

Sarah Mei
@sarahmei

14 May

The inebriation test reminds me that everything in my code should provide information, _beyond_ the semantics of what it does.

Sarah Mei
@sarahmei

Follow

Variable names should indicate intent. Structure should make the possible execution paths obvious, & the likely ones even _more_ obvious.

2:50 AM - 14 May 2016

19 32

3.2 During the Evaluation Period, you hereby agree that the Licensed Software is provided "AS IS" with no representation, guarantee or warranty of any kind as to its functionality, quality, performance, suitability or fitness for purpose. All other terms, conditions, representations and warranties expressed or implied whether by statute or otherwise are hereby expressly excluded.

3.3 We shall not be liable for any claim, damages or other liability arising from or in connection with your use of the Licensed Software during the Evaluation Period.

3.4 For the avoidance of doubt, during the Evaluation Period: (a) clauses 5.1, 8.1, 9.3 and 13.2 of this Agreement shall not apply; and (b) clause 13.1 shall apply except that the reference to clause 13.2 is deleted.

3.5 Before or upon expiry of the Evaluation Period:

3.5.1 if, in your sole opinion, the Licensed Software has met your requirements, and you wish to continue to use the Licensed Software beyond the end of the Evaluation Period, you can decide whether to obtain a licence to the Free Edition Software or the equivalent Licence Fee version. Once the appropriate the licence has been obtained, this Agreement shall continue in force (except that this clause 3 shall no longer apply).

3.5.2 if you decide that the Licensed Software does not meet your requirements, or otherwise do not wish to enter into a paid up Licence, then you shall destroy the Licensed Software and all copies, in any form including partial copies or modifications of the Licensed Software received from us or made in connection with this Licence and all documentation relating thereto. Any rights of yours to use the Licensed Software shall cease.

4 OWNERSHIP OF INTELLECTUAL PROPERTY RIGHTS

4.1 You acknowledge that:

4.1.1 all Intellectual Property Rights in or relating to the Licensed Software are owned by or licensed to us or licensed to us for business use; and

```
private string _getLocalDbInstanceNameFromServerString(string databaseServer, bool startLocalDbInstanceIfNotRunning)
{
    if (IsSqlLocalDb) // Support for Sql Server LocalDB connections, denoted by the data source string, ie "(localdb)\MyInstanceName". Retrieves the Named Pipes connection string, necessary for our logic
    {
        if (!databaseServer.Contains(@"\"))
            throw new SqlConnectionException(this, "Cannot lookup info on Sql Server LocalDB. The string '" + databaseServer + "' does not contain an instance name.");

        var sqlLocalDbCmdPath = _getSqlLocalDbLocation();
        if (sqlLocalDbCmdPath == null)
            throw new SqlConnectionException(this, "Cannot connect to local instance because the SqlLocalDB utility could not be found. To fix this, please install SQL Server Data Tools from microsoft.c

        var instanceInfo = CommandExecutor.ExecReturnOutput(sqlLocalDbCmdPath, "info " + InstanceName);

        if (!instanceInfo.Contains("\nState:              "))
            throw new SqlConnectionException(this, "Cannot connect to Sql Server LocalDB: " + instanceInfo);

        if (!instanceInfo.Contains(LocalDbStateRunning))
        {
            if (startLocalDbInstanceIfNotRunning)
            {
                Log.Warn("The instance '" + databaseServer + "' is not running, will attempt to start. Additional info: \n" + instanceInfo);

                var startOutput = CommandExecutor.ExecReturnOutput(sqlLocalDbCmdPath, "start " + InstanceName);
                instanceInfo = CommandExecutor.ExecReturnOutput(sqlLocalDbCmdPath, "info " + InstanceName);

                if (!instanceInfo.Contains(LocalDbStateRunning))
                    throw new SqlConnectionException(this, "Could not start the Sql Server LocalDB instance '" + databaseServer + "': " + startOutput);
            }
            else
            {
                throw new SqlConnectionException(this, "Cannot connect to Sql Server LocalDB. The instance '" + databaseServer + "' is not running. Additional info: \n" + instanceInfo);
            }
        }

        var namedPipesPos = instanceInfo.IndexOf(LocalDbNamedPipes.Label);
        if (namedPipesPos == -1)
            throw new SqlConnectionException(this, "Cannot connect to Sql Server LocalDB. Cannot locate Named Pipes identifier in instance '" + databaseServer + "': " + instanceInfo);

        var namedPipesValuePos = namedPipesPos + LocalDbNamedPipes.Label.Length;
        var namedPipesEndPos = instanceInfo.IndexOf("\n", namedPipesValuePos);
        var namedPipesString = namedPipesEndPos == -1 ? instanceInfo.Substring(namedPipesValuePos) : instanceInfo.Substring(namedPipesValuePos, namedPipesEndPos - namedPipesValuePos);
        return namedPipesString.Trim();
    }
    return databaseServer;
}
```

The Four Rules of Simple Design

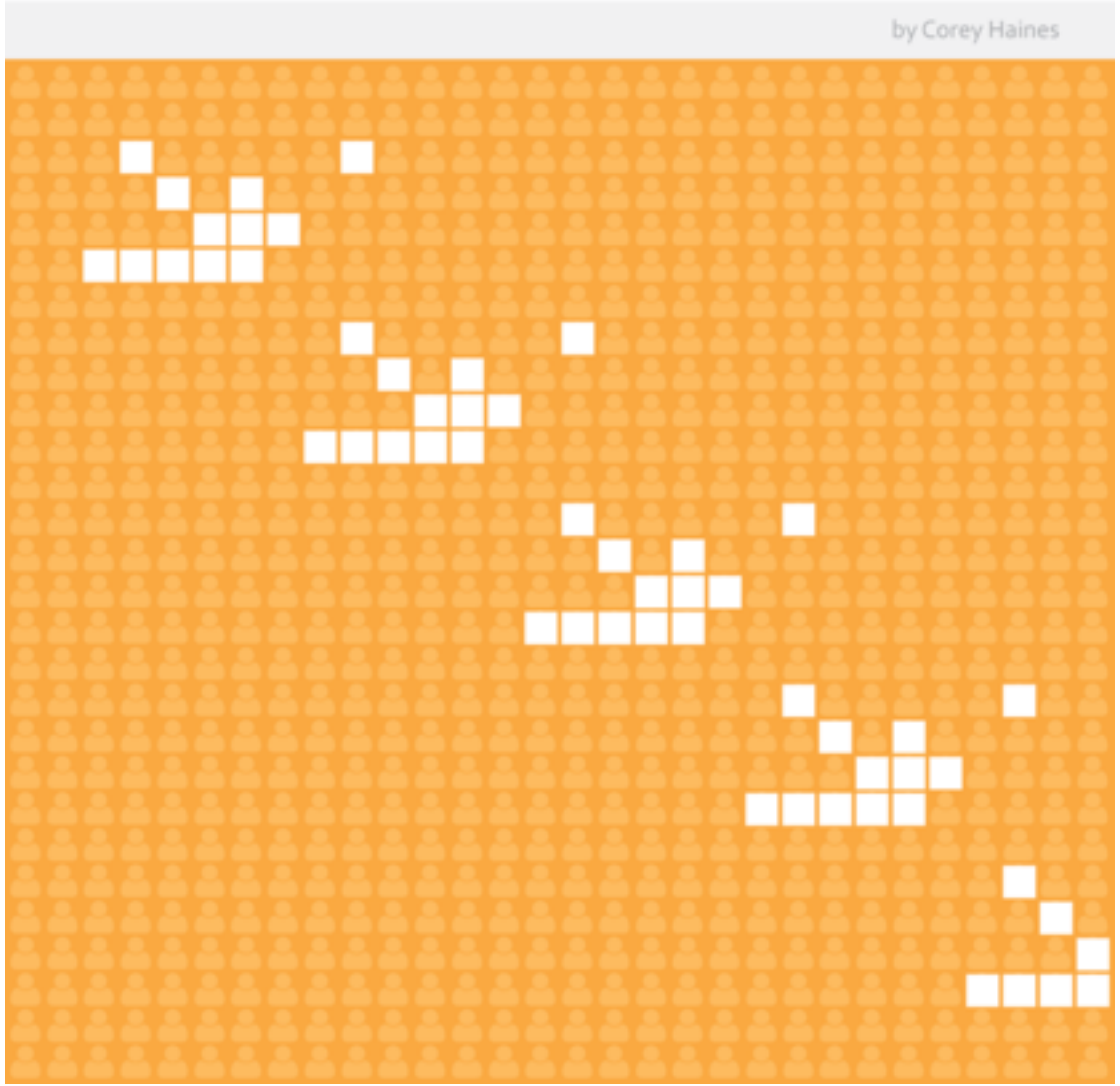
1. Tests Pass
2. Expresses Intent
3. No Duplication (DRY)
4. Small

UNDERSTANDING THE

4 RULES OF SIMPLE DESIGN

And other lessons from watching 1000's of pairs work on Conway's Game of Life

by Corey Haines



Improving your tests

Test names should influence your API design

Root out subtle duplication

Test names influence API design

```
public void A_New_Carriage_Is_Empty()
{
    Assert.That(new Carriage().IsEmpty() == true);
    Assert.That(new Carriage().Passengers.Count == 0);
}

public void A_Seat_Can_Be_Assigned()
{
    var carriage = new Carriage();
    carriage.AssignSeat(34);
    Assert.That(carriage.IsSeatAssigned(34) == true);
    Assert.That(carriage.AssignedSeats == 1);
}
public void After_Assigning_A_Seat_The_Carriage_Is_Not_Empty()
{
    var carriage = new Carriage();

    carriage.AssignSeat(34);

    Assert.That(carriage.IsEmpty() == false);
}
```

Root out subtle duplication

```
public void A_Seat_Can_Be_Assigned()
{
    var carriage = new Carriage();
    var seat = Seat.Number(34);
    carriage.AssignSeat(34);
    carriage.Assign(seat);
    Assert.That(carriage.IsSeatAssigned(34) == true);
}   Assert.That(carriage.IsAssigned(seat) == true);
}
```

Improving your objects

Avoid procedural polymorphism

Invert composition to break inheritance

Avoid procedural polymorphism

```
public class EmptyCarriage : Carriage
{
    public override bool IsAcceptingReservations()
    {
        return IsOpenForBookings();
    }
}

public class Carriage
{
    public virtual bool IsAcceptingReservations()
    {
        if (IsEmpty)
            return IsOpenForBookings();
        return HasSpacesRemaining();
    }
}
```

Anti-IF Campaign

HOME | GET STARTED | JOIN THE CAMPAIGN | EXPERIENCES | ABOUT

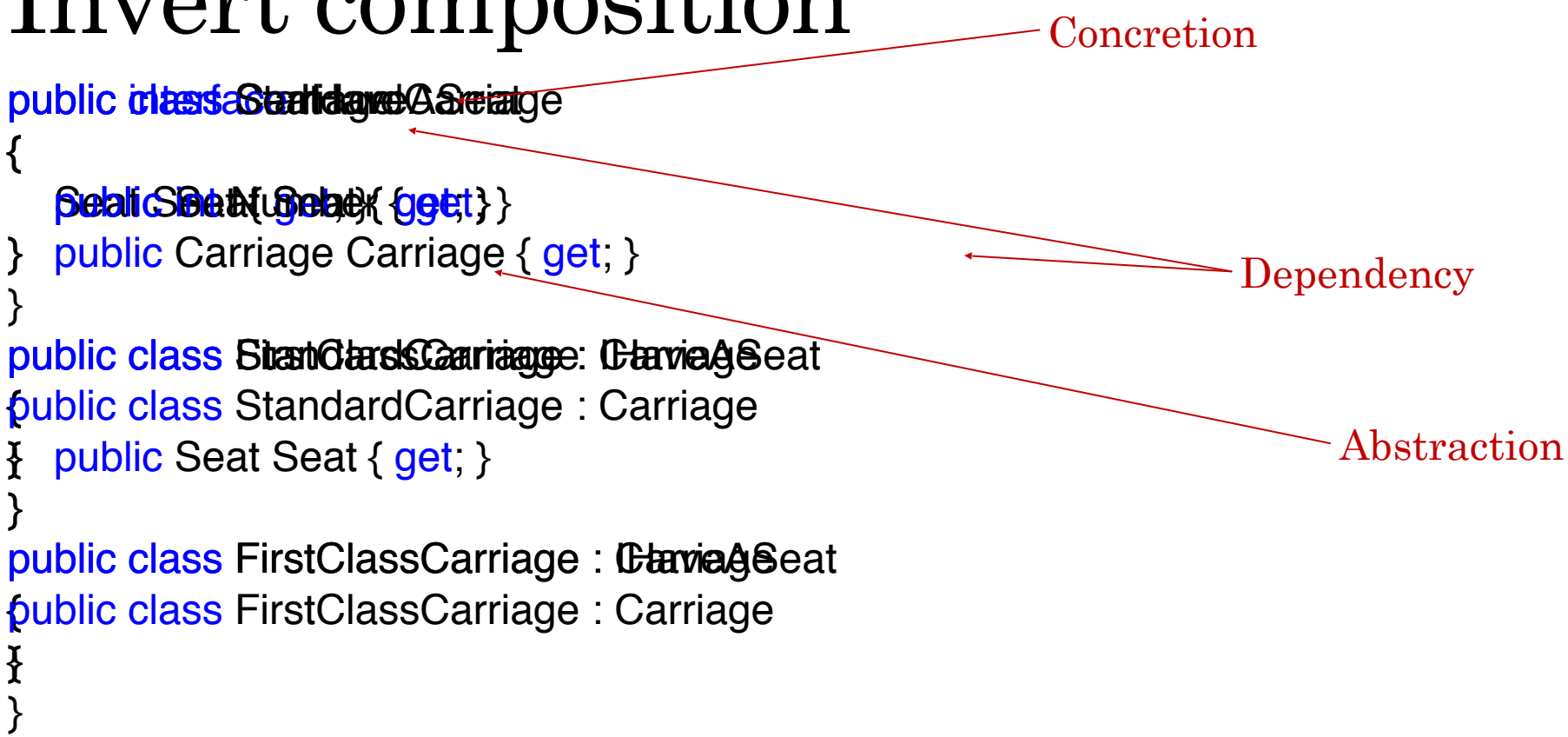
Less IFs, more power.

Have you ever wondered how IFs impact on your code? Avoid dangerous IFs and use Objects to build a code that is flexible, changeable and easily testable, and will help avoid a lot of headaches and weekends spent debugging! Share how to write effective code the easy way!

The goal of the Anti-IF Campaign is to raise awareness of the effective use of software design principles and practices, by first of all removing bad, dangerous IFs.

```
//Bond class
double calculateValue() {
      if(_type == BTP)  {
return calculateBTPValue();
} else if (_type == BOT) {
return calcalateBOTValue();}
else {
return calculateEUBValue();
}
}
```

Invert composition



Sandi Metz
@sandimetz

Follow

Don't write code that guesses the future, arrange code so you can adapt to the future when it arrives.

5:59 PM - 5 Mar 2014

359 177

“You’ll never know less than you know right now”

– *Kent Beck*

Express Yourself!

Alastair Smith ~ I T.A.K.E. Unconference 2016
@alastairs ~ alastair@alastairsmith.me.uk