File | New Project

Choose “Class Library”

Name the project “Cat”

Choose the location to save the file.

Name the solution “ResharperDemo3”

Talk about the right margin bar

Red = Error

Yellow = Warning

Green = Hint

Blue = TODO

Hover over the yellow bar. Double click on the yellow bar.

Introduce first short cuts

Shift+ALT+PgDn to go to the next error

Shift+ALT+PgUp to go to the previous error

Alt+PgDn to go to the next error, warning or suggestion

Alt+PgUp to go to the previous error, warning or suggestion

Introduce Alt+Enter and the red light bulb.

Remove the unused references

Rename the class (Ctrl+R, R), Type “Cat” talk about the suggested names.

Note: the class is renamed, and the file is renamed.

Note: the green squiggly line.

Create the private backing fields for Name

Note: as you type Resharper limits the choices according to the letters type (camel casing is supported also)

Note: the \_name is grayed out. Resharper detected that is unused code and could be removed.

Create the string Property for Name.

Note: The \_name property is no longer grey.

Introduce the first refactor. Move to the name property (Alt+Up Arrow)

Alt+Enter to see the ReSharper Quick Fixes

Introduce ReSharper templates

Type prop, Notice the Yellow Square, hit Tab

String as TYPE and Age as name.

Create Speak Method (public void Speak())

Type “Console.”: Notice the System.Console import suggestion. Hit Alt+Enter, this only works if the project is referenced.

Type WL: Notice the Camel Case intellisense filtering.

After WriteLine is available notice how the intellisense changes based on the text that you type.

Type out Cat speaking: Meow”, CTRL+SHIFT+ENTER. This will fill in the parenthesis and the semi colon.

Save Cat.cs

# Create the Interface

Now a new requirement comes in and we need to have a dog and another animal called, ItsAReallyLongAnimal so we decide to create an Animal interface called IAnimal.

Jump back up to the Cat class. Alt+\ is a shortcut for jump to a member; you can type the member name or scroll up or down.

Now that you are at the Cat class, let’s extract the interface. CTRL+SHIFT+R, brings up the Refactor this menu. Type X to Extract to Interface…

Up comes a dialog box. Choose the name IAnimal, then which members you want to move.

Move the IAnimal interface to its own class file.

First we need to jump to that Interface, CTRL+T (go to type), type IAnimal

Next, move the interface: CTRL+R, o

Note: there is a new file.

Let’s rename the namespace

Navigate to the namespace Cat, change it to Animal

Navigate to the class cat, note it has changed to Animal.

Change the project name and assembly properties (no way to do this in Resharper)

Save everything (CTRL+S)

# Create the Dog

Jump to the Cat class (Ctrl+T), then Cat

Copy Type. CTRL+SHIFT+R, C. Type Dog.

New class and file are created, let’s change the speak method.

Move to the Speak method. ALT+\, then S enter;

# Create the ThisIsAReallyLongName animal

Jump to the Dog class (Alt+\), Dog.

Copy Type. CTRL+SHIFT+R, C. Type ThisIsAReallyLongName.

New class and file are created, let’s change the speak method.

Move to the Speak method. ALT+\, then S enter;

# Add XML Comments to the Interface

Jump to the interface (CTRL+T), then type IA.

Alt+Enter (Add Comment Stub)

With GhostDoc (Free addin to Visual Studio at <http://www.submain.com>) CTRL+SHIFT+D

Add in all of the XML Comments. Notice the file is green.

Update the comments on the Cat.

Ctrl+T, Cat

Alt+Enter, Add Comment Stub

Alt+Down Arrow twice to get to the Name property

Alt+Enter, C for Copy comments from base.

Repeat for Age and Speak

Update the comments on the Dog.

Ctrl+T, Dog

Alt+Enter, Add Comment Stub

Alt+Down Arrow twice to get to the Name property

Alt+Enter, C for Copy comments from base.

Repeat for Age and Speak

Update the comments on the ThisIsAReallyLongAnimal.

Ctrl+T, TIA

Alt+Enter, Add Comment Stub

Alt+Down Arrow twice to get to the Name property

Alt+Enter, C for Copy comments from base.

Repeat for Age and Speak

Save all files.

# Create the Test Project

File | New Project…, choose Test Project (or class library if using nUnit or other testing framework)

Call it AnimalTest

Add to Solution

Add a reference to the Animal class library.

In the empty class file, add the TestFixture attribute. Press Alt+Enter to import the using statement for NUnit.Framework.

Rename the class to DogTest

Add the [Test] Attribute.

Type public void TestAge Ctrl+Shift+Enter

Type int age=5;

Type var dog = new Dog( notice that ReSharper enters DogTest, you will have go back a delete Test to import Animal.Dog if you have not already imported it.

Type Dog.Age = age;

Notice the read curly braces. The Age is defined as string but we are trying to assign it an int. Let’s fix it.

Ctrl+F12 takes us to the implementation of the Age property. F12 takes us to the definition of the property. Alt+Home will take us to where we can change the property, the base.

Change the signature, Ctrl+R, S, change to Int32 (ReSharper does not support the int keyword shortcut). Click next and the value is changed to an int everywhere.

Go back to DogTest (Ctrl+T), then type DT.

Note the red curly underlining is gone.

Note there is a green underline under the new keyword.

Click on the new keyword and then Alt+Enter,U for Use object Initializer.

Type Assert.That(name, Is.EqualTo(dog.Name));

Create the TestName method

[Test]

public void TestName()

{

string name = "Spike";

var dog = new Dog {Name = name};

Assert.That(name, Is.EqualTo(dog.Name));

}

Check out the Quick Do feature (Ctrl+Shift+F1)

Clean up the code.

Run the Unit Tester.

## Templatize the test.

Select the contents of the class and copy to clipboard.

Choose ReSharper | Live Templates

Click on File Templates

Expand User Templates

Click C#

Click Add new template (Yellow square with green plus)

Change the default file extension to Test.cs

Change the Available in to C#

Paste the text (formatting is not that import, ReSharper will reformat on using the template)

Replace parts that can change like the test name with $*variableName*$

Change the Dog with $Animal$, change dog with animal.

Save.

Click on the Animal Test project

Choose ReSharper | New From Template | More …

Expand User Templates

Click on Animal Test.

Optionally add it to the quick list (this will make it available on the ReSharper menu)

Click Ok

Give it the name of CatTest

Type Cat and the test case are done.

Append the test to the session

Create the new test for ThisIsAReallyLongName.

Choose ReSharper | New From Template | Animal Test

Type ThisIsAReallyLongName press enter

Type ThisIsAReallyLongName press tab

Append the test to the session

Run all of the test

# Create the WinForms Application

File | New Project …

Choose Windows Forms Application

Name the project AnimalApplication

Add a reference to the Animal library

Double click on the Form to create the Form1\_Load event

Type List<Cat> c, notices it will pluralize cat to cats for you.

cats.Add(createCat("Sammie", 5));

Navigate back to createCat. Alt+Enter C, Create Method ‘createCat’

createCat should look like this

private Cat createCat(string name, int age)

{

return new Cat {Name = name, Age = age};

}

# Code Modifying Shortcuts

### Move Code Blocks

Ctrl+Alt+Shift+Up Arrow: moves the code item up

Ctrl+Alt+Shift+Down Arrow: moves the code item down

Ctrl+Alt+Shift+Left Arrow: moves the code item left

Ctrl+Alt+Shift+Right Arrow: moves the code item right

## Duplicate line

Ctrl+D

## Commenting

Ctrl+Shift+/: Block Comment

Ctrl+Alt+/: Comment

# Navigation Shortcuts

Navigate from Anywhere

Go to symbol (Shift+Alt+T)

Go to file (Ctrl+Shift+T)

Go to file member (Alt +\)

## Navigate from occurrence of a specific symbol

Go to declaration (F12)

Go to type declaration (Ctrl+Shift+F12)

Go to base (Alt+Home)

Go to inheritor (Alt+End)

Go to Usages (Shift+Alt+F12)

## Navigate from anywhere within a member

Go to Next Member/Tag (Alt+Down)

Go to Previous Member / Tag (Alt+Up)

Go to Containing Definition (Ctrl + [)

## Other Windows

Find Window (Ctrl+Shift+Alt+F12)

File Structure (Ctrl+Alt+F)

Type Hierarchy (Ctrl+Alt+H)

Navigate from here (Alt+`)

Highlight usages (Shift+Alt+F11)

Todo explorer (Ctrl+Alt+D)

Stack Trace Explorer (Ctrl+E, T)

Recent Files (Ctrl++)