

forget about the new

@tomkuj

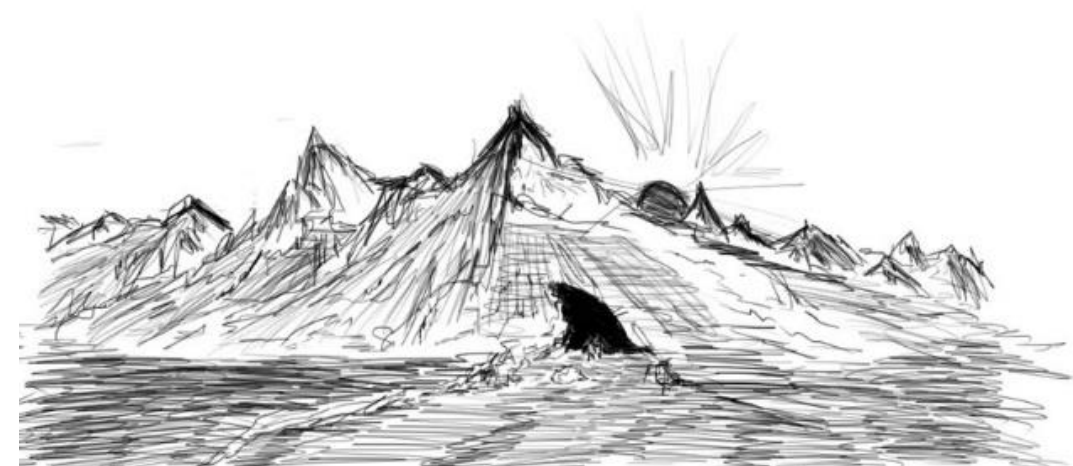


Part of the GFT Group

you are going to stop using new() operator

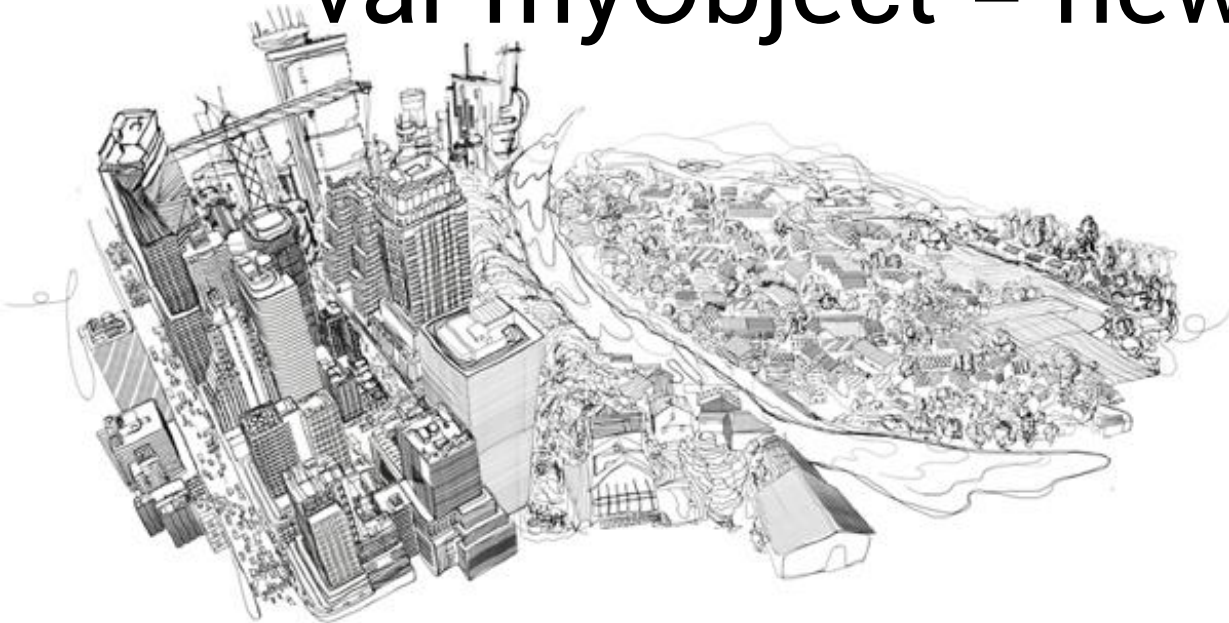
how the world looks now?

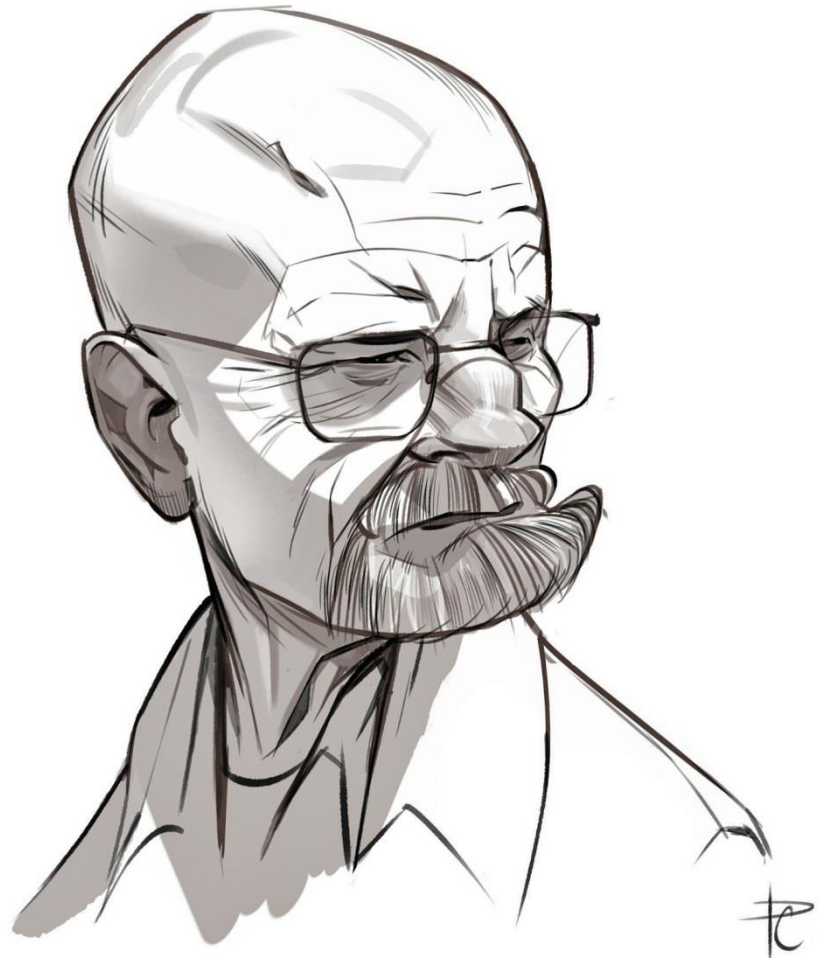
object myObject = new object();



how the world looks now?

```
var myObject = new MyClass(obj1, obj2);
```





why is it so bad?

managing the dependencies between objects



```
public class MyClass
{
    public MyClass(SqlConnection connection, OtherObject obj)
    {
        ///...
    }
}
```

<http://tinyurl.com/roslyn-method-compiler>

▼ Src

▶ .nuget

▼ Compilers

▶ Core

▼ CSharp

▶ csc

▶ csc2

▶ Desktop

▼ Portable

▶ Binder

▶ BoundTree

▶ CodeGen

▶ Compilation

▼ Compiler

AnonymousTypeMethod

ClsComplianceChecker

Compiler.cs

DocumentationComment

DocumentationComment

DocumentationComment

EntryPointCandidateFinder

MethodBodySynthesizer

MethodBodySynthesizer

MethodCompiler.cs

MethodCompiler.cs

Compare with other versions: ↔

```
}

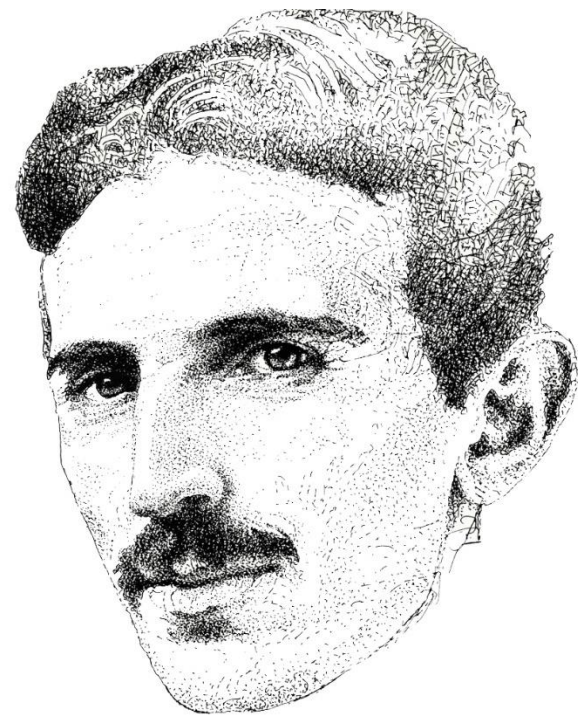
public static void CompileMethodBodies(
    CSharpCompilation compilation,
    PEModuleBuilder moduleBeingBuiltOpt,
    bool generateDebugInfo,
    bool hasDeclarationErrors,
    DiagnosticBag diagnostics,
    Predicate<Symbol> filterOpt,
    CancellationToken cancellationToken)
{
    Debug.Assert(compilation != null);
    Debug.Assert(diagnostics != null);

    using (Logger.LogBlock(FunctionId.CSharp_Compiler_CompileMethodBodies, message: compilation.PreviousSubmission != null))
    {
        if (compilation.PreviousSubmission != null)
        {
            // In case there is a previous submission, we should ensure
            // it has already created anonymous type/delegates templates

            // NOTE: if there are any errors, we will pick up what was created anyway
            compilation.PreviousSubmission.EnsureAnonymousTypeTemplates(cancellationToken)

            // TODO: revise to use a loop instead of a recursion
        }
    }
}
```

hard to test



how to **avoid** that?

Inversion of Control

Dependency Injection

```
public class MyClass
{
    public MyClass(IConnection connection, IOtherObject obj)
    {
        ///...
    }
}
```



<http://autofac.org/>

0_greeter.zip

1. Run it. This is simple application.
2. Check the configuration file –
GreeterApp.exe.config
3. There are two implementations of **IAnswerer** provided:
 - 1.PolishAnswerer**
 - 2.EnglishAnswerer**
4. Stop application, change the interface in a proper place – rerun.



```
<autofac>
  <components>
    <component type="GreeterApp.PolishAnswerer, GreeterApp"
               service="GreeterApp.IAnswerer, GreeterApp" />
  </components>
</autofac>
```


1_writer.zip

Print the actual time.

REQUIREMENTS:

- different outputs + colors + other?
- time in different formats



your task:

<http://tinyurl.com/ioc-1-writer>



easy to test

loose coupling

easy to change objects impl

q&a