# Relation among classes

## Agenda

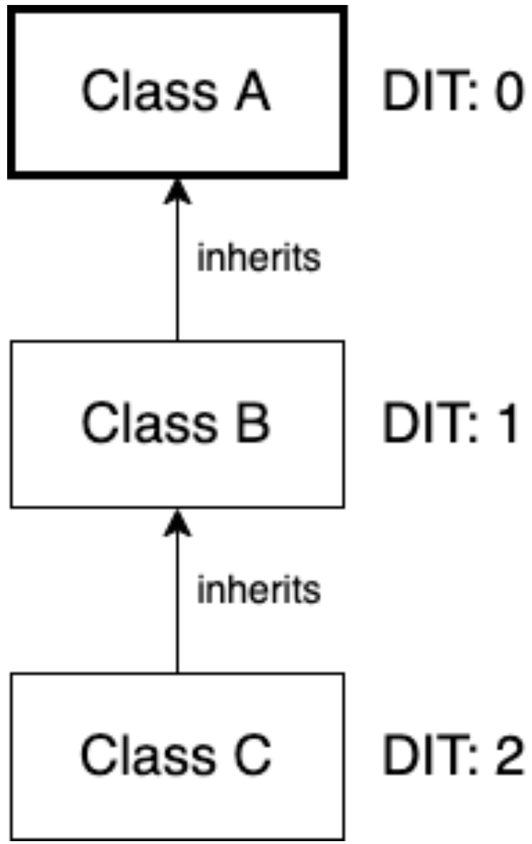
- Depth of the inheritance tree (DIT)
- Number of children (NOC)
- Method inheritance factor (MIF)
- Response for a class (RFC)

# Depth of the inheritance tree (DIT)

# Depth of the inheritance tree (DIT)

## What is it? How to Calculate?

- Measurement for class tree complexity
- The depth of a class in the inheritance tree is the maximum length from the node to the root of the tree.
- The higher the DIT the higher the level of complexity and the potential for errors and code reuse



# Depth of the inheritance tree (DIT)

## Software for automation?

- Built-in feature in Visual Studio, but neither Visual Studio Mac nor Visual Studio Code

# Number of children (NOC)

# Number of children (NOC)

## What is it?

- The metric after **Number OF Children(NOC)** is part of the group of the measures, which are based on inheritance hierarchies
- indicator of the potential influence a class can have on the design and on the system
- NOC is the number of direct specializations of a class
- The more complex the inheritance hierarchy – e.g. the larger the Number of Children (NOC) - the more important it is to test the deriving class intensively

# Number of children (NOC)

## How to calculate?

- Calculate the number of immediate subclasses subordinate to a class in the hierarchy
- Consult your Inheritance Tree
- Count the number of children that are one level below the class which is under calculation

# Method inheritance factor

# Method inheritance factor (MIF)

## What is it?

$$\text{METHOD INHERITANCE FACTOR} = \frac{\text{sum of the inherited methods in all classes of the system}}{\text{total number of available methods for all classes}}$$

# Method inheritance factor (MIF)

## How to calculate?

$$MIF = \frac{\sum_{i=1}^{TC} M_i(C_i)}{\sum_{i=1}^{TC} M_a(C_i)}$$

where:

$$M_a(C_i) = M_d(C_i) + M_i(C_i)$$

and:

$M_d(C_i)$ = the number of methods declare in a class

$M_a(C_i)$ = the number of methods that can be invoked in association with $Ci$

$M_i(C_i)$ = the number of methods inherited (and not overriden in $Ci$).

$TC$ = Total number of classes in the system under consideration.

# Method inheritance factor (MIF)

## How to calculate?

```
class Vehicle {
  public void honk() {
    System.out.println("Tuut, tuut!");
  }
}

class Car extends Vehicle {
  private String modelName = "Mustang";

  public void start() {
    System.out.println("Brrr, brrr");
  }
}

class Fahrrad extends Vehicle {
  public void speicheverbiegtsich() {
    System.out.println("üüüühhhht");
  }
}
```

| | all | inherited |
|---|---|---|
| vehicle | 1 | 0 |
| car | 2 | 1 |
| Fahrrad | 2 | 1 |
| MIF | 2 : 5 = 0.4 | |

# Response for a class (RFC)

# Response for a class (RFC)

## What is it?

- The **Response for Class (RFC)** metric is the total number of methods that can potentially be executed in response to a message received by an object of a class.
- This number is the sum of the methods and constructors of a class.
- Inherited methods are counted, but overridden methods are not.
- High score means a lot of methods, harder to read, understand and test the code.

# Response for a class (RFC)
## How to calculate?

- Addition of:
  - Number of Class’ Methods
  - Number of invoked Methods

# Response for a class (RFC)

## How to calculate?

```
ClassMethod CreateProjection(cls As %String, ByRef params) As %Status
{
    set ns=$namespace
    new $namespace
    znspace "%SYS"

    if ('##class(Security.Applications).Exists(..#CSPAPP)) {
        do ##class(Security.System).GetInstallationSecuritySetting(.security)
        set cspProperties("AutheEnabled") = $select((security="None"):64,1:32)
        set cspProperties("NameSpace") = ns
        set cspProperties("Description") = ..#CSPAPPDESCRIPTION
        set cspProperties("DispatchClass") = ..#ROUTER
        write !, "Creating WEB application """_..#CSPAPP_"""..."
         $$$ThrowOnError(##class(Security.Applications).Create(..#CSPAPP, .cspProperties))
        write !, "WEB application """_..#CSPAPP_""" created."

        if ##class(%Studio.General).GetWebServerPort(,,,.url) {
             write !, "You can now open it with a link: "_url_$p(..#CSPAPP,"/",2,*)_"/"
        }
    } else {
        write !, "WEB application """_..#CSPAPP_""" already exists, so it is ready to use."
    }
     Quit $$$OK
}
```

Example: RFC is 5

# Response for a class (RFC)
## Software for automation?

- Source code based metric, automation for a compiled product difficult
- Extensions like ‘objectscriptQuality’ for IDEs like VSCode
- Some Code Linters can highlight this