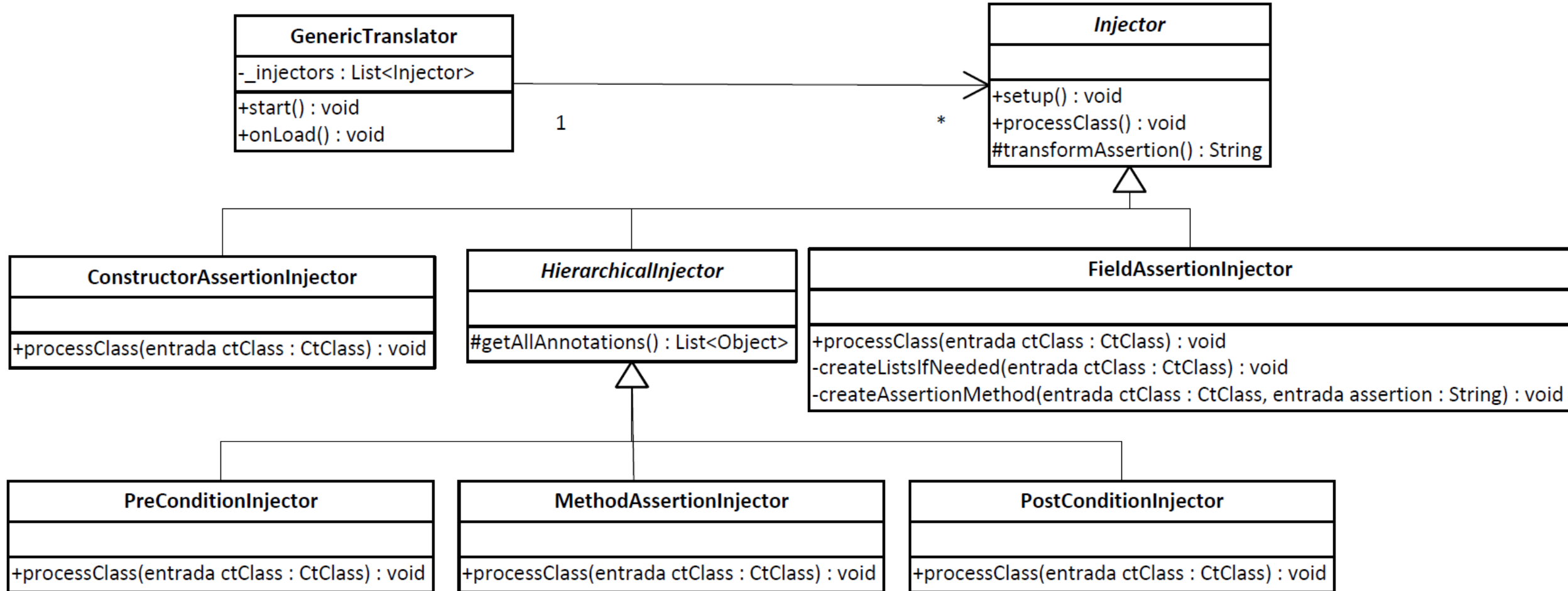


Improving the Java Type System



Daniel Cardoso 66964
Francisco Raposo 66986
Miguel Roxo 66259

System Design



Field Initialization

Create lists if not already created in topmost class //hierarchically

For each behavior //method or constructor

 For each asserted field access

 If read access

 Inject code before access to check if field is initialized

//in list

 If write access

 Inject code after access to mark field as initialized //add

it to list

- 2 lists are created (per topmost class):
 - static list for static fields
 - non-static list for non-static fields

Field Assertions

For each behavior //method or constructor

For each asserted field access

Store field assertions in list

If write access

For each assertion

Transform assertion string

Inject code after access to check for transformed

assertion

- 2 types of assertion transformation can happen:
 - replacement of <field name> for <class name>.<field name> (static fields)
 - creation of temporary method in corresponding class containing the original assertion (non-static fields)

Method Assertions

For each method

 Store assertions (including overridden methods) in a list

 For each assertion

 For each argument

 Create and initialize backup variable

 Replace argument for backup names in assertion code

 Inject modified assertion code before method return

- Assertion list is sorted from most generic to most specific class

Extensions

- Constructor Assertions
- Methods Pre/Post Conditions

Constructor Assertions

For each constructor

 Store assertions in a list

 For each assertion

 Inject assertion code on method entry

- Does not go through the class hierarchy

Preconditions

For each method

 Store preconditions (including overridden methods) in a list

 For each precondition

 Inject precondition code before on method entry

- Greater programming flexibility (not limited to method return assertions)

Postconditions

For each method

Store postconditions (including overridden methods) in a list

For each postcondition

For each argument

Create and initialize backup variable

Replace argument for backup names in postcondition code

Inject modified postcondition code before method return

- The same as method assertions