THE UNIVERSITY OF MELBOURNE

SWEN20003 Workshop

Week 09

Demonstrator: Xulin Yang

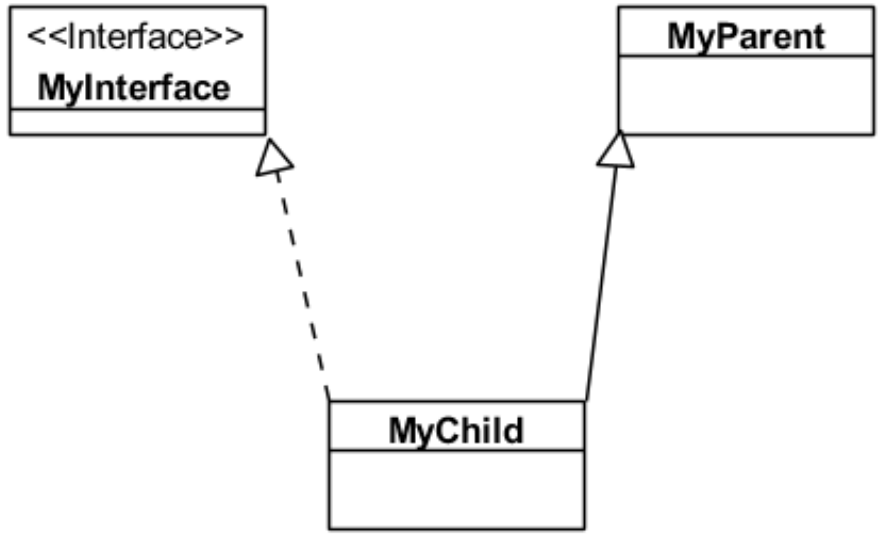
Workshop tasks

1. About project 1
2. Go through questions
3. Divide groups by records
4. Do questions Q1, 2, 3, 4, 5, 6, 7 together
5. Take attendance

Emphasis for Week 08

White arrow Black line for extend (inherit) parent class

White arrow Dotted line for implement (realize) interface



About Project 1

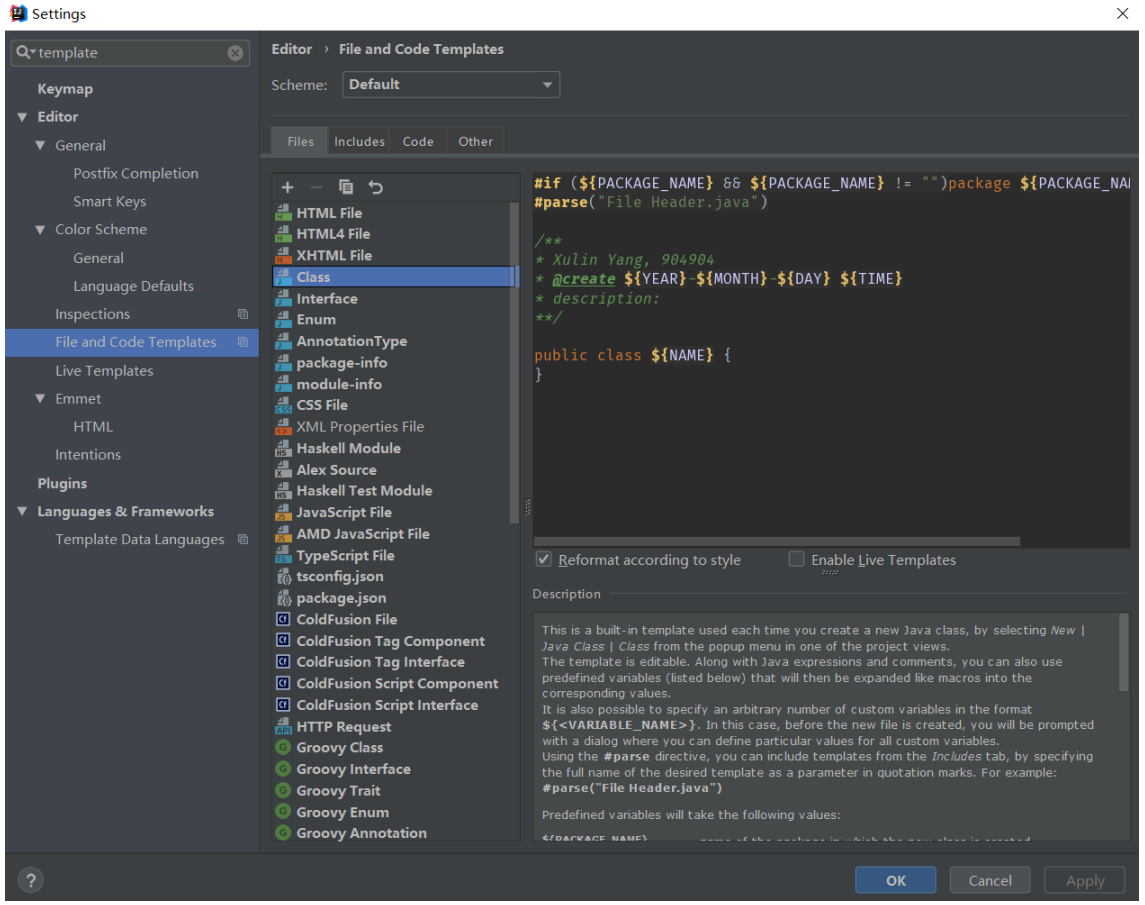
1. I mark my 2 tutorials' Delegation Use of methods Cohesion Code style criteria.
2. If you have question about the feedback, you can send email to head tutor + me and explains your concern.
3. Please take a look at the sample solution.
4. What is good commenting?
   A. File level documentation
   B. Class documentation
   C. Javadoc for attribute + method

Census for my marked students	Delegation	Use of methods	Cohesion	Code style	Sum
average	0.880952381	0.964285714	0.77380952	0.928571429	3.547619
median	1	1	1	1	3.5

About Project 1 – File level documentation

Configure file template for your java class.

author + student id + create time + description



About Project 1 - Class documentation

If your class is complicated {

left click your class name;

Alt + Enter;

Add Javadoc to describe what your class does, etc

}

About Project 1 - Javadoc for attribute + method

For attributes, each of them should has Javadoc; or one line for several obvious and similar attributes

For class method (constructor, public/private method), add Javadoc (@param if any, @return if any, a description)

For getter & setter, group them together and use one line comment says “here are the getter and setter for this class”

1. left click your attribute/method name;
2. Alt + Enter;
3. Add Javadoc to describe what it does, etc

Generic Programming

What are the advantages of using generically typed classes and methods?

What are the advantages of using generically typed classes and methods?

Generic programming is more modular

It allows you to write algorithms and data structures that can operate on/with any class, instead of having to repeatedly write the same code for any class you may wish to use.

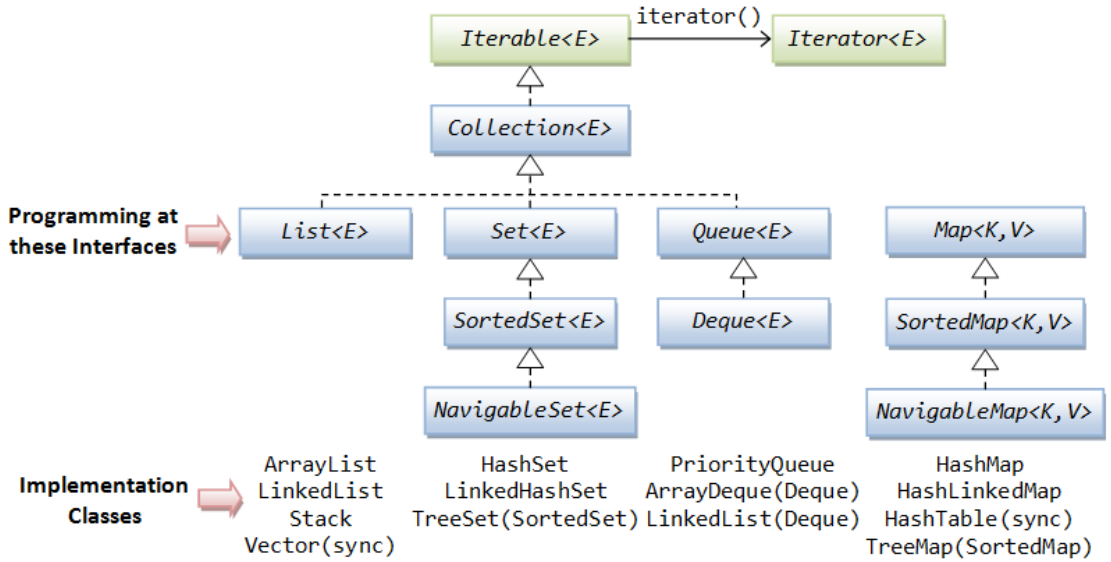
Give some examples of where we would write our own generic classes, because the inbuilt Java generic classes are insufficient

Give some examples of where we would write our own generic classes, because the inbuilt Java generic classes are insufficient

Java's generic classes are predominantly for dealing with collections of objects, either in the form of its Collection framework, a Map, or sorting. Applications outside of this require us to write our own classes.

The first instance is where we wish to extend the functionality of the existing classes; for example we may wish to extend the functionality of a List object to allow us to operate on the data inside it.

The second instance is where the desired functionality doesn't exist at all, and we need to create it; for example, defining a Pair class to bind two objects together. (But actually, you have the built-in Pair class)



THE UNIVERSITY OF MELBOURNE

Thank you