Lab 10: Generics

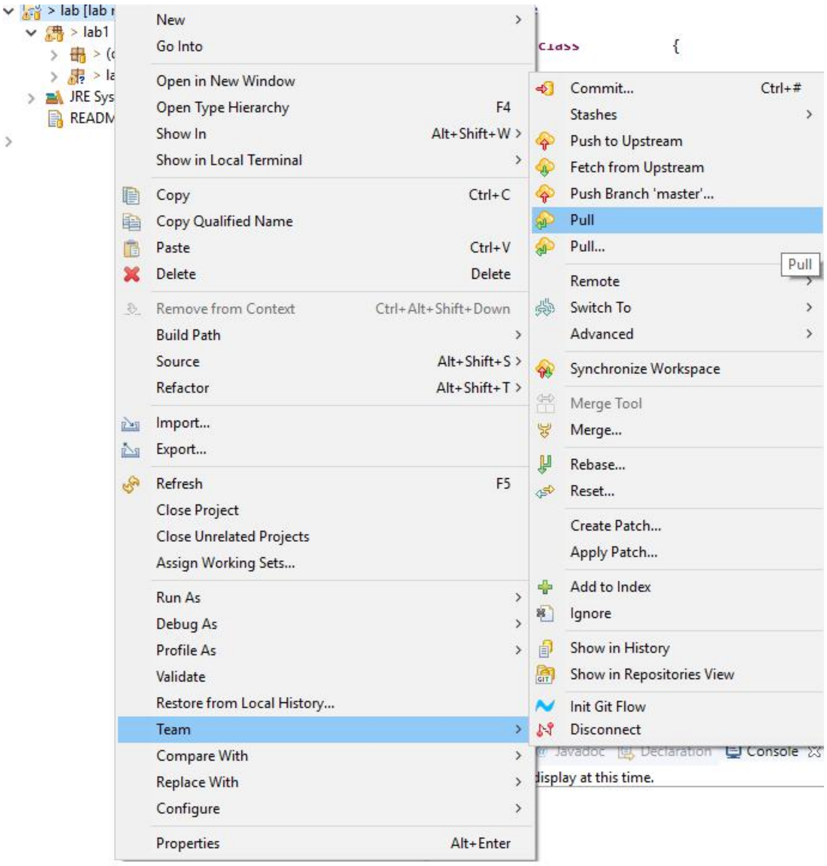
In this lab we will be using generics in implementing an abstract data type called stack[1].

Setup

1. Merge your repository in order get the contents of “lab10” from OzgurKilic/lab.and “stack” as described in previous lab documents.

2. In command line, change directory to lab directory which was created after cloning your repository. If you haven’t clonned your repository yet, you should clone it as described in previous labs.

3. If you are using eclipse click the “Pull” item in the “Team” menu



4. If you are not using eclipse, execute the following command in the lab directory.
   a) git pull
   b) Change directory to lab10

Exercise 1 : Implement the Stack Interface using ArrayList class

1. Define a class named StackArrayListImpl implementing the Stack interface.
2. Implement the behaviours defined in Stack interface defined in this class
3. Create a test class and test your implementation

Exercise 2 : Modify the Stack Interface and the StackArrayListImpl to support Generics

1. To update the Stack interface to use generics, you create a generic type declaration by changing the code "public interface Stack" to "public interface Stack<T>". This introduces the type variable, T, that can be used anywhere inside the interface.
2. Update the StackArrayListImpl class to support generics.
3. Test your interface and class.

Exercise 3 : Implement the stack interface wihout using any collection classes or array structure

1. Define a class named StackImpl implementing the Stack interface.
2. Define a StackItem class.
3. Stack item should contain the value instance variable to hold a value in a stack and should reference item that is below it.
4. Test your StackImpl class.

Exercise 4 : Wildcards

1. Add the following method to Stack interface to extract content of a stack to a List.

   ```
   public List<T> toList();
   ```

2. Add the following method to Stack interface to support adding contents of another stack to the current stack,

   ```
   public void addAll(Stack<T> aStack);
   ```

3. Implement the methods in classes implementing the stack interface
4. Test the addAll method by creating two stack instances and add one to another
5. Try to add Stack<String> to Stack<Object>
6. Use Wildcards to handle step5

NOTE: Your lab will **not be graded** if

- Your account name does not have the format described in lab1.pdf
- Your repository name is not lab
- Your files have compilation errors
- You haven’t complete the steps described in exercises
- Your added/modified files are not submitted to Bitbucket.
    - You have to add commit and push files as described in lab1.pdf

[1] http://www.wiki-zero.com/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvU3RhY2tfKGFic3RyYWN0X2RhdGFfdHlwZSk