https://github.com/gomezem/classwork/tree/master/workspaceCSCI4830-ex-0312a-gomez

🥘 workspaceCSCl4830-ex-0303-gomez - simple-refactoring-examples/src/q1/extract_method/refactored/A.java - Eclipse IDI <u>File Edit Source Refactor Navigate Search Project Run Window Help</u> | 🗂 ▼ 🔚 🕼 | 🎔 💋 >> 📵 📵 T | 📵 | 🙍 | № III III | ■ 18 | № 2. Ф. 10 | ₹ 😿 | 卷 ▼ 💽 ▼ 💁 ▼ 👺 🕳 🖋 Project Ex... 🛭 🗀 ☑ A.java ☑ A.java
☒ 1 package q1.extract method.refactored; 2 3 import java.util.List; > Mark JRE System Library 🗸 🕭 src 5 public class A { q1.extract_meth Node m1(List<Node> nodes, String p) { 6⊜ > 🗾 A.java 7 extractedMethod(nodes, p); 8 return null; > 🗾 A.java 9 10 11⊖ Edge m2(List<Edge> edgeList, String p) { A.java extractedMethod(edgeList, p); 12 √ # q2.form_templa 13 return null; A.java 14 } 15 16⊜ <T extends Graph> void extractedMethod (List<T> objs, String p) { 17 for (T obj: objs) { if (obj.contains(p)) 18 19 System.out.println(obj); 20 } 21 } 22 } 23 24 **class** Graph { 25 String name; 26⊜ boolean contains(String p) { 27 return name.contains(p); 28 29 } 30 31 class Node extends Graph { 32 String name; 33 **△**34⊝ public boolean contains(String p) { 35 return name.contains(p); 36 37 } 38 20 aless Edge symbols Charb

```
workspaceCSCI4830-ex-0303-gomez - simple-refactoring-examples/src/q2/form_template_method1/refactored/Ajava - Eclipse IDE Elle Edit Source Refactor Navigate Search Project Bun Window Help
```

```
Project Ex... 🛭 🗀 🗖
                              🖟 A.java 🖟 A.java 🖺 A.java 🛭
       E $ 7 | 9 8
                                1 package q2.form_template_method1.org;
                                                                                                                                        1 package q2.form_template_method1.refactored;

→ B simple-refactoring-ex

                                2
3 public class A {
40 public static void main(String[] args) {
    → MRE System Library
                                                                                                                                         public class A {
public static void main(String[] args) {
                                            System.out.println("Before");

System.out.println("m1:" + (new A().m1(0, 3)));

System.out.println("m2:" + (new A().m2(0, 3)));
                                                                                                                                                   System.out.println("After");
System.out.println("m1:" + (new A().m1(0, 3)));
System.out.println("m2:" + (new A().m2(0, 3)));

y 

⊕ q1.extract meth

          > 🗓 A.java
                             int m1(int min, int max) {
    int sum = 0;
    for (int i = min; i <= max; i++) {
        sum += i;
    }
    return sum;
}</pre>
       v # q1.extract_meth
                                                                                                                                     100 int m1(int min, int max) {
11    return new M1().common(min, max);
12  }
13
          > 🗓 A.iava
      v ∰ q2.form_templa 10®
          > 🗓 A.java
       > 🗓 A.java
                                                                                                                                             return new M2().common(min, max);
}
                                                                                                                                      149 int m2(int min, int max) {
                                                                                                                                      15
16
17 }
                               17
18® in
19
20
21
22
23
24 }
25 }
                                      int m2(int min, int max) {
  int sum = 0;
  for (int i = min; i <= max; i++) {
    sum += i * i;</pre>
                                                                                                                                      18
19 abstract class Template {
                                                                                                                                             rouseract class !emplate {
  int common(int min, int max) {
   int sum = 0;
   // TODO: Replace the FOR statement below with your answer.
   for (int i = min; i <= max; i++) {
      sum += apply(i);
   }
}</pre>
                                                                                                                                      20⊖
21
                                                                                                                                      23
24
25
                                            return sum:
                                                                                                                                                  return sum;
                                                                                                                                      29
30 }
31
                                                                                                                                              abstract int apply(int i);
                                                                                                                                      32 class M1 extends Template {
                                                                                                                                              int apply(int i) {
                                                                                                                                    △35
```

Emily Gomez 03/12-a

```
1 package q1.extract_method.refactored;
                                                                                                                package q2.form_template_method1.refactored;
   3 import java.util.List;
                                                                                                                 public class A {
                                                                                                                    public static void main(String[] args) {
                                                                                                                        System.out.println("After");
System.out.println("m1:" + (new A().m1(0, 3)));
System.out.println("m2:" + (new A().m2(0, 3)));
   5 public class A {
           Node m1(List<Node> nodes, String p) {
               extractedMethod(nodes, p);
               return null;
  10
                                                                                                             100
                                                                                                                    int m1(int min, int max) {
   return new M1().common(min, max);
          Edge m2(List<Edge> edgeList, String p) {
    extractedMethod(edgeList, p);
 11⊝
                                                                                                             11
                                                                                                             12
               return null;
                                                                                                             13
                                                                                                                    int m2(int min, int max) {
   return new M2().common(min, max);
 14
          }
                                                                                                             149
                                                                                                             15
  15
  16⊜
           <T extends Graph> void extractedMethod (List<T> objs, String p) {
                                                                                                             16
               for (T obj: objs) {
   if (obj.contains(p))
                                                                                                             17 }
  17
  18
                                                                                                             18
                         System.out.println(obj);
                                                                                                             19 abstract class Template {
  20
              }
                                                                                                             20⊝
                                                                                                                   int common(int min, int max) {
   int sum - 0:
     < 1
  21
                                                                                                                                                                    m × % 🗎 🔝
🖺 Markers 🗔 Properties 🦚 Servers 🗯 Data Source Explorer 🔓 Snippets 📮 Console 🛛
<terminated> A (1) [Java Application] C:\Program Files\Java\jre1.8.0_231\bin\javaw.exe (Mar 31, 2020, 9:38:15 PM)
After
m1:6
m2:14
```

Running after:

