Emily Gomez
03/12-a

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

workspaceCSCI4830-ex-0303-gomez - simple-refactoring-examples/src/q1/extract_method/refactored/A.java - Eclipse ID

File Edit Source Refactor Navigate Search Project Run Window Help

Project Ex...

- simple-refactoring-ex
  - JRE System Library
  - src
    - q1.extract_meth
      - A.java
    - q1.extract_meth
      - A.java
    - q2.form_templa
      - A.java
    - q2.form_templa
      - A.java

A.java | A.java

```java
package q1.extract_method.refactored;

import java.util.List;

public class A {
    Node m1(List<Node> nodes, String p) {
        extractedMethod(nodes, p);
        return null;
    }

    Edge m2(List<Edge> edgeList, String p) {
        extractedMethod(edgeList, p);
        return null;
    }

    <T extends Graph> void extractedMethod (List<T> objs, String p) {
        for (T obj: objs) {
            if (obj.contains(p))
                System.out.println(obj);
        }
    }
}

class Graph {
    String name;
    boolean contains(String p) {
        return name.contains(p);
    }
}

class Node extends Graph {
    String name;

    public boolean contains(String p) {
        return name.contains(p);
    }
}

class Edge extends Graph {
```

Emily Gomez
03/12-a

workspaceCSCI4830-ex-0303-gomez - simple-refactoring-examples/src/q2/form_template_method1/refactored/A.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Ex...
- simple-refactoring-ex
  - JRE System Library
  - src
    - q1.extract_metl
      - A.java
    - q1.extract_metl
      - A.java
    - q2.form_templa
      - A.java
    - q2.form_templa
      - A.java

A.java | A.java | A.java

```java
package q2.form_template_method1.org;

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)));
   }

   int m1(int min, int max) {
      int sum = 0;
      for (int i = min; i <= max; i++) {
         sum += i;
      }
      return sum;
   }

   int m2(int min, int max) {
      int sum = 0;
      for (int i = min; i <= max; i++) {
         sum += i * i;
      }
      return sum;
   }
}
```

A.java

```java
package q2.form_template_method1.refactored;

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)));
   }

   int m1(int min, int max) {
      return new M1().common(min, max);
   }

   int m2(int min, int max) {
      return new M2().common(min, max);
   }
}

abstract class Template {
   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);
       }
      return sum;
   }

   abstract int apply(int i);
}

class M1 extends Template {

   @Override
   int apply(int i) {
      return i;
   }
}
```

```java
class M2 extends Template {

   @Override
   int apply(int i) {
      return i * i;
   }
}
```

Emily Gomez
03/12-a

A.java

```java
package q1.extract_method.refactored;

import java.util.List;

public class A {
    Node m1(List<Node> nodes, String p) {
        extractedMethod(nodes, p);
        return null;
    }

    Edge m2(List<Edge> edgeList, String p) {
        extractedMethod(edgeList, p);
        return null;
    }

    <T extends Graph> void extractedMethod (List<T> objs, String p) {
        for (T obj: objs) {
            if (obj.contains(p))
                System.out.println(obj);
        }
    }
```

A.java

```java
package q2.form_template_method1.refactored;

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)));
    }

    int m1(int min, int max) {
        return new M1().common(min, max);
    }

    int m2(int min, int max) {
        return new M2().common(min, max);
    }
}

abstract class Template {
    int common(int min, int max) {
        int sum = 0;
```

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: