Xuedinan Gao

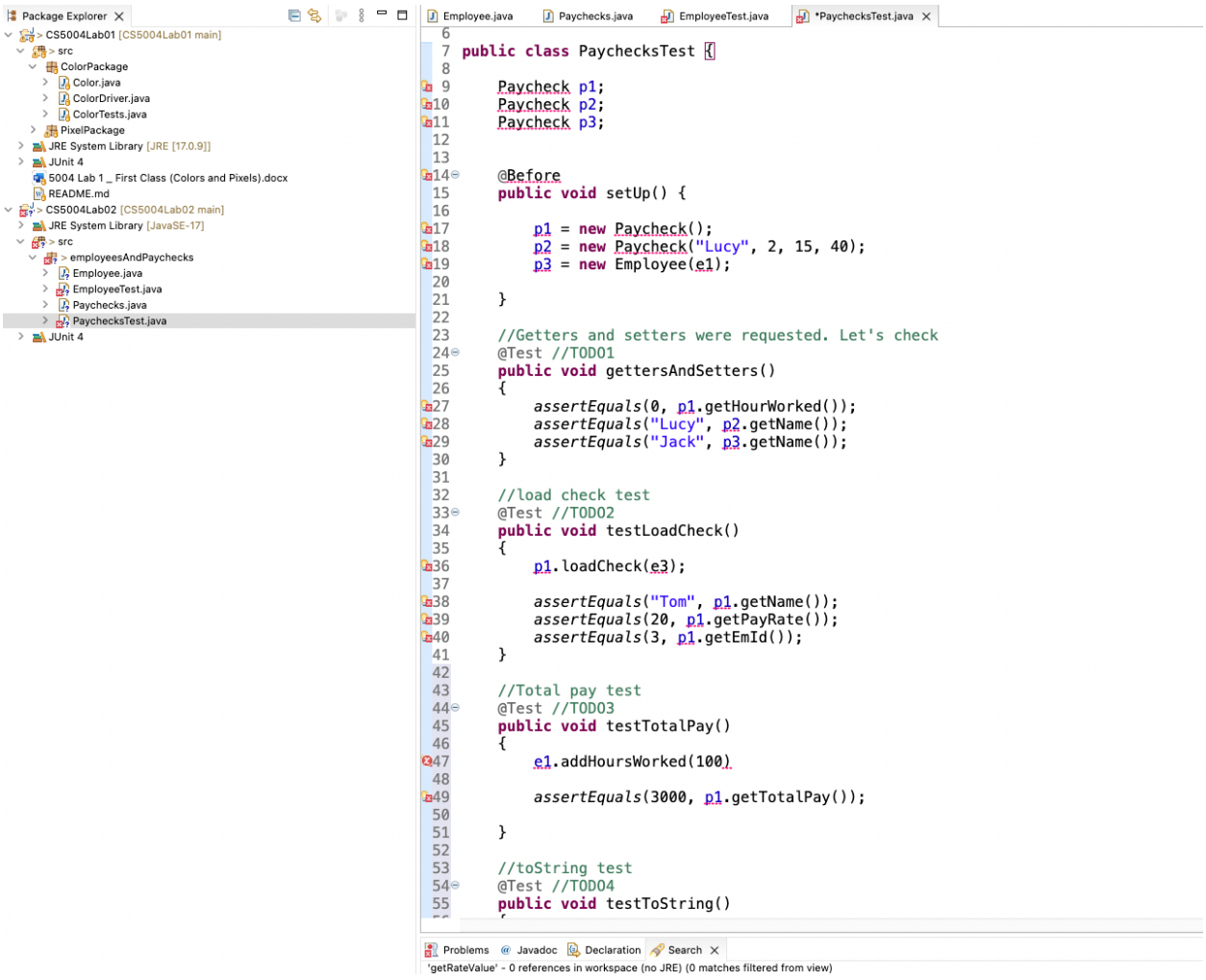
1/23/2024

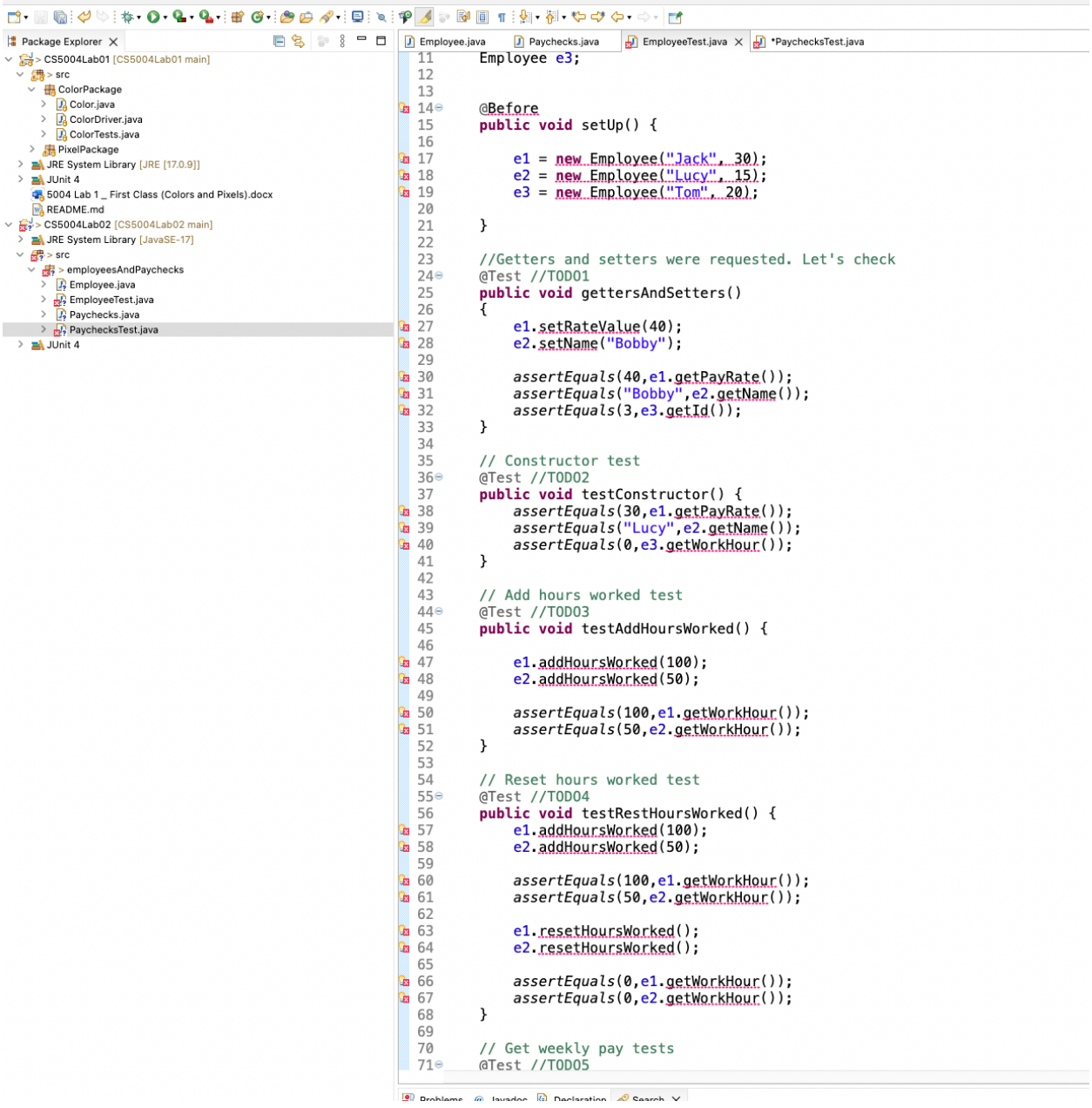
⭐ Weekly report

This week, I learned the three major topics, Override method (like toString()), how to use static variables when design class, and what's the exception handling and the use case for it.

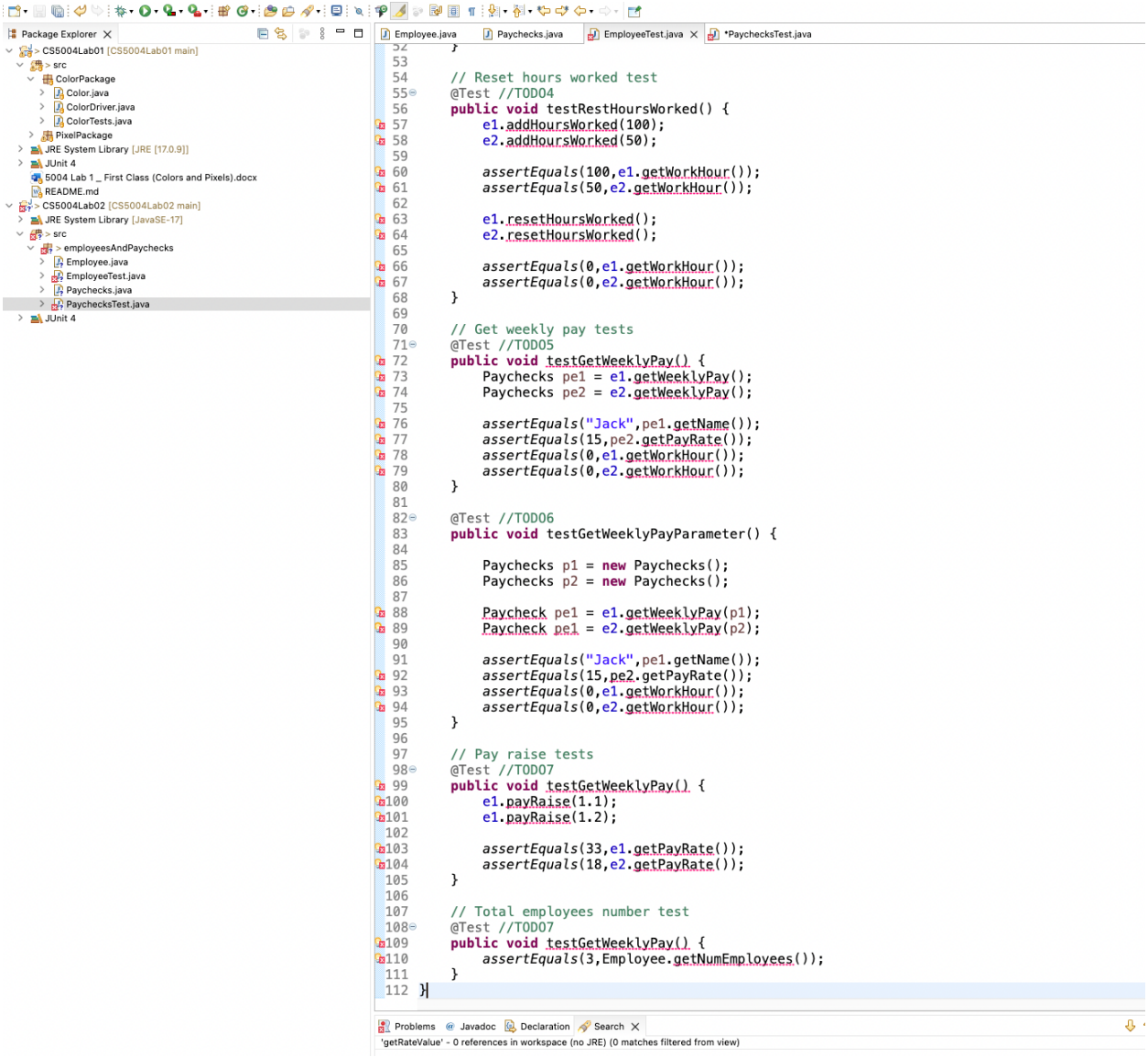
For the override method, it's more like we extend a class with certain method, and we are able to re-write method based on our own application. For the static, it's scope of a variable or method that we could use it to make them be usable when call the class by itself, not just call it only by creating objective before. For the exception handling, it's a smart way and option to make our program not crash, especially when there are uncertain cases we can't 100% control, like open a file, like ask user input some content.

For the following extensions, I'm expecting to have more introduction videos for the concept of designing of a program. And I'd like to learn a strategic way to design classes with manipulation of using public, private, static, final, exception, etc.. Also I'm glad to have more practices along with code demo videos, that's really helpful content for study.


Package Explorer
> CS5004Lab01 [CS5004Lab01 main]
> src
ColorPackage
Color.java
ColorDriver.java
ColorTests.java
PixelPackage
JRE System Library [JRE [17.0.9]]
JUnit 4
5004 Lab 1 _ First Class (Colors and Pixels).docx
README.md
> CS5004Lab02 [CS5004Lab02 main]
JRE System Library [JavaSE-17]
> src
> employeesAndPaychecks
Employee.java
EmployeeTest.java
Paychecks.java
PaychecksTest.java
JUnit 4
Employee.java
Paychecks.java
EmployeeTest.java
*PaychecksTest.java
6
7 public class PaychecksTest {
8
9 Paycheck p1;
10 Paycheck p2;
11 Paycheck p3;
12
13
14 @Before
15 public void setUp() {
16
17 p1 = new Paycheck();
18 p2 = new Paycheck("Lucy", 2, 15, 40);
19 p3 = new Employee(e1);
20
21 }
22
23 //Getters and setters were requested. Let's check
24 @Test //TODO1
25 public void gettersAndSetters()
26 {
27 assertEquals(0, p1.getHourWorked());
28 assertEquals("Lucy", p2.getName());
29 assertEquals("Jack", p3.getName());
30 }
31
32 //load check test
33 @Test //TODO2
34 public void testLoadCheck()
35 {
36 p1.loadCheck(e3);
37
38 assertEquals("Tom", p1.getName());
39 assertEquals(20, p1.getPayRate());
40 assertEquals(3, p1.getEmId());
41 }
42
43 //Total pay test
44 @Test //TODO3
45 public void testTotalPay()
46 {
47 e1.addHoursWorked(100)
48
49 assertEquals(3000, p1.getTotalPay());
50
51 }
52
53 //toString test
54 @Test //TODO4
55 public void testToString()
Problems
Javadoc
Declaration
Search
'getRateValue' - 0 references in workspace (no JRE) (0 matches filtered from view)




```
11      Employee e3;
12
13
14      @Before
15      public void setUp() {
16
17          e1 = new Employee("Jack", 30);
18          e2 = new Employee("Lucy", 15);
19          e3 = new Employee("Tom", 20);
20
21      }
22
23      //Getters and setters were requested. Let's check
24      @Test //TODO1
25      public void gettersAndSetters()
26      {
27          e1.setRateValue(40);
28          e2.setName("Bobby");
29
30          assertEquals(40,e1.getPayRate());
31          assertEquals("Bobby",e2.getName());
32          assertEquals(3,e3.getId());
33      }
34
35      // Constructor test
36      @Test //TODO2
37      public void testConstructor() {
38          assertEquals(30,e1.getPayRate());
39          assertEquals("Lucy",e2.getName());
40          assertEquals(0,e3.getWorkHour());
41      }
42
43      // Add hours worked test
44      @Test //TODO3
45      public void testAddHoursWorked() {
46
47          e1.addHoursWorked(100);
48          e2.addHoursWorked(50);
49
50          assertEquals(100,e1.getWorkHour());
51          assertEquals(50,e2.getWorkHour());
52      }
53
54      // Reset hours worked test
55      @Test //TODO4
56      public void testRestHoursWorked() {
57          e1.addHoursWorked(100);
58          e2.addHoursWorked(50);
59
60          assertEquals(100,e1.getWorkHour());
61          assertEquals(50,e2.getWorkHour());
62
63          e1.resetHoursWorked();
64          e2.resetHoursWorked();
65
66          assertEquals(0,e1.getWorkHour());
67          assertEquals(0,e2.getWorkHour());
68      }
69
70      // Get weekly pay tests
71      @Test //TODO5
```


Package Explorer
CS5004Lab01 [CS5004Lab01 main]
src
ColorPackage
Color.java
ColorDriver.java
ColorTests.java
PixelPackage
JRE System Library [JRE [17.0.9]]
JUnit 4
5004 Lab 1 _ First Class (Colors and Pixels).docx
README.md
CS5004Lab02 [CS5004Lab02 main]
JRE System Library [JavaSE-17]
src
employeesAndPaychecks
Employee.java
EmployeeTest.java
Paychecks.java
PaychecksTest.java
JUnit 4
Employee.java
Paychecks.java
EmployeeTest.java
*PaychecksTest.java
52 }
53
54 // Reset hours worked test
55 @Test //TODO4
56 public void testRestHoursWorked() {
57 e1.addHoursWorked(100);
58 e2.addHoursWorked(50);
59
60 assertEquals(100,e1.getWorkHour());
61 assertEquals(50,e2.getWorkHour());
62
63 e1.resetHoursWorked();
64 e2.resetHoursWorked();
65
66 assertEquals(0,e1.getWorkHour());
67 assertEquals(0,e2.getWorkHour());
68 }
69
70 // Get weekly pay tests
71 @Test //TODO5
72 public void testGetWeeklyPay() {
73 Paychecks pe1 = e1.getWeeklyPay();
74 Paychecks pe2 = e2.getWeeklyPay();
75
76 assertEquals("Jack",pe1.getName());
77 assertEquals(15,pe2.getPayRate());
78 assertEquals(0,e1.getWorkHour());
79 assertEquals(0,e2.getWorkHour());
80 }
81
82 @Test //TODO6
83 public void testGetWeeklyPayParameter() {
84
85 Paychecks p1 = new Paychecks();
86 Paychecks p2 = new Paychecks();
87
88 Paycheck pe1 = e1.getWeeklyPay(p1);
89 Paycheck pe1 = e2.getWeeklyPay(p2);
90
91 assertEquals("Jack",pe1.getName());
92 assertEquals(15,pe2.getPayRate());
93 assertEquals(0,e1.getWorkHour());
94 assertEquals(0,e2.getWorkHour());
95 }
96
97 // Pay raise tests
98 @Test //TODO7
99 public void testGetWeeklyPay() {
100 e1.payRaise(1.1);
101 e1.payRaise(1.2);
102
103 assertEquals(33,e1.getPayRate());
104 assertEquals(18,e2.getPayRate());
105 }
106
107 // Total employees number test
108 @Test //TODO7
109 public void testGetWeeklyPay() {
110 assertEquals(3,Employee.getNumEmployees());
111 }
112 }
Problems
Javadoc
Declaration
Search
'getRateValue' - 0 references in workspace (no JRE) (0 matches filtered from view)


Run
Debug
Stop
Share
Save
{ } Beautify
Main.java
Widget.java
WidgetFactory.java
Requirements.txt
FacConstants.java
Widget w5 = null;
Widget w6 = null;
try{
wf1.addResource(-100);
w5 = wf1.makeWidget();
w6 = wf1.makeWidget();
}
catch(IllegalArgumentException IAE){
System.out.println(IAE);
System.out.println("Adding 100 resource");
wf1.addResource(100);
}
catch(Exception E){
System.out.println(E);
}
System.out.println(w5);
System.out.println(w6);
*/
//Part 3 : Widget Equality
/*
System.out.println(w1 == w3); //what does this really mean?
//The unpredictability of == with objects
String s1 = "cat";
String s2 = "cat";
String s3 = new String("cats");
System.out.println(s1.equals(s2));
System.out.println(s1.equals(s3));
System.out.println();
*/
Widget w8 = new Widget(100);
System.out.println(w8.getWidget());
WidgetFactory wf9 = new WidgetFactory();
try{
wf9.addResource(110);
Widget w9 = new Widget();
w9 = wf9.makeWidget();
System.out.println(w9.getWidget());
System.out.println(w8.equals(w9));
}
catch(IllegalArgumentException IAE){
System.out.println(IAE);
}
}
}


Run
Debug
Stop
Share
Save
{ } Beautify
Main.java
Widget.java
WidgetFactory.java
Requirements.txt
FacConstants.java
/*
The amazing Widget
Author : Dr. G
Date : 9/22/23
*/
class Widget
{
private static int counter = 0;
private int id;
private int weight;
public Widget(){
id = counter;
counter++;
weight = FacConstants.WIDGIT_WEIGHT;
}
public Widget(int weight){
id = counter;
counter++;
this.weight = weight;
}
public Widget getCopy(){
return new Widget();
}
public int getWidget(){ return weight;}
@Override
public String toString(){ return String.valueOf(id); }
public static int thing(){ return 999; }
}

Run | Debug | Stop | Share | Save | { } Beautify

Main.java | Widget.java | WidgetFactory.java | Requirements.txt | FacConstants.java

```java
/*
The amazing Widget

Author : Dr. G
Date : 9/22/23

*/

public class WidgetFactory{


    private int resourceBin;

    public WidgetFactory(){
        resourceBin = 0;
    }

    public void addResource(int resource)throws IllegalArgumentException{

        if(resource < 0){
            throw new IllegalArgumentException("Attempted to send negative resource amount");
        }

        resourceBin += resource;
    }

    public Widget makeWidget()throws Exception{

        if((resourceBin - FacConstants.WIDGIT_WEIGHT) < 0){

            throw new Exception("Not enough resource to produce widget;");
        }

        resourceBin -= FacConstants.WIDGIT_WEIGHT;

        return new Widget(resourceBin);
    }

    public static Widget makeWidgets(int amount)throws Exception{

        if(amount < 0){
            throw new IllegalArgumentException("Attempted to send negative resource amount");
        }

        if((amount - FacConstants.WIDGIT_WEIGHT) < 0){

            throw new Exception("Not enough resource to produce widget;");
        }


        return new Widget();
    }
}
```

I understand that my learning is dependent on individual effort and struggle, and I acknowledge that this assignment is a 100% original work and that I received no other assistance other than what is listed here.

Acknowledgements and assistance received:

I did not use generative AI in any form to create this content and the final content was not adapted from generative AI created content.

I did not view content from any one else's submission including submissions from previous semesters nor am I submitting someone else's previous work in part or in whole.

I am the only creator for this content. All sections are my work and no one else's with the exception being any starter content provided by the instructor. If asked to explain any part of this content, I will be able to.

By putting your name and date here you acknowledge that all of the above is true and you acknowledge that lying on this form is a violation of academic integrity and will result in no credit on this assignment and possible further repercussions as determined by the Khoury Academic Integrity Committee.

Name: Xuedinan Gao	Date: 1/23/2024