**Homework 2**

1. (10 pts) What are the diagrams defined in the UML Standard. Give a one or two sentence description of each one.

Structural Diagram – Represent the static component of the system, which represents portions of the diagram that make up for the main structure. Types of structural diagrams include class diagrams, object diagrams, component diagrams, and deployment diagrams.

Behavioral Diagrams – represent any system that can have two components, static and dynamic. These types of diagrams encapsulate the dynamic aspect of the system. Kinds of behavioral diagrams include Use case, sequence, collaboration, statechart, and activity diagrams.  
  
2. (10 pts) Given the following code, how should the toString methods in the classes H2ClassA and H2ClassB be written to give the indicated output and take advantage of the natural toString method in H2ClassB?

1  import java.util.ArrayList;  
2   
3  public class H2ClassA {  
4    ArrayList <H2ClassB> list = new ArrayList <H2ClassB> ();  
5   
6    public static void main (String args []) {  
7      H2ClassA y = new H2ClassA ();  
8      int [] v = {4, 3, 7, 5, 99, 3};  
9      for (int m: v)   
10       y.list.add (new H2ClassB (m));  
11     System.out.println (y);  
12   } // end main  
13   
14 } // end class H2ClassA  
15   
16 class H2ClassB {  
17   int x;  
18   H2ClassB (int a) { x = a;}  
19 } // end H2ClassB

OUTPUT:

4 3 7 5 99 3

The toString method should be written as follows:

For H2A

public String toString() {

String str;

For(i = 0; i< list.length();i++) {

Str += list.get(i).toString() + “ “;

}

return str;

}

For H2B

Public String toString(){

return a + “”;

}

3. (10 pts) How can the following code be corrected? Give at least two good answers.

1 public class H2ClassC {  
2   H2ClassC (int a) {}  
3 } // end class H2ClassC  
4   
5 class H2ClassD extends H2ClassC{  
6 } // end class H2ClassD

There is no default constructor in 'H2ClassC'. A constructor must be created to match its superclass.

4. (10 pts) Why does the following code give a compiler error? How should it be fixed?

1  public class H2ClassE {  
2    int x, y, z;  
3   
4    H2ClassE (int a) {  
5      x = a;  
6      this (5, 12);  
7    }  
8   
9    H2ClassE (int b, int c) {  
10     y = b;  
11     z = c;  
12   }  
13 } // end class H2ClassE

Call to 'this()' (line 6) must be first statement in constructor body instead of after x = a.

5. (10 pts) What is wrong with the following declaration? How should it be fixed?

public static final int myNumber = 17.36;

the value that was assigned to myNumber is a double type due to being a decimal (2 places after decimal point. Either make this value an actual integer or cast this decimal value to int.

6. (10 pts) What is wrong with the following code? How should it be fixed?

1 public class H2ClassG {  
2   final int x;  
3   
4   H2ClassG () {}  
5   H2ClassG (int a) {x = a;}  
6 } // end class H2ClassG

x is defined as a constant and must instead be defined as an int.

7. (10 pts) What is wrong with the following code? How should it be fixed?

1 public class H2ClassH {  
2   final int x;  
3   
4   int H2ClassH () {  
5     if (x == 7) return 1;  
6     return 2;  
7   } // end  
8 } // end class H2ClassH

This code needs a constructor with an int of x inside of the parameters, initializing x to itself.

**public class** H2ClassH {  
 **final int x**;  
  
 **public** H2ClassH(**int** x) {  
 **this**.**x** = x;  
 }  
  
 **int** H2ClassH () {  
 **if** (**x** == 7) **return** 1;  
 **return** 2;  
 } *// end  
//*} *// end class H2ClassH*

8. (10 pts) What is wrong with the following code? x should be given a value of 24. What are two ways this can be legally accomplished?

1 public class H2ClassI {  
2   final int x;  
3   
4   public static void main (String args []) {  
5     H2ClassI h = new H2ClassI ();  
6     h.x = 24;  
7   } // end main  
8 } // end class H2ClassI

This code needs a constructor with int x in it, initializing x to itself. Also, x should be declared as a static int. otherwise, it will not work in the code.

**public class** H2ClassI {  
 **static int** *x*;  
  
 **public** H2ClassI(**int** x) {  
 **this**.*x* = x;  
 }  
  
 **public static void** main (String args []) {  
 H2ClassI h = **new** H2ClassI (*x*);  
 h.*x* = 24;  
 } *// end main*} *// end class H2ClassI*

9. (10 pts) What is wrong with the following code? Give two effective ways to fix it.

1  import javax.swing.\*;  
2  import java.awt.event.\*;  
3   
4  public class H2ClassJ extends JFrame {  
5    public static final long serialVersionUID = 22;  
6   
7    public H2ClassJ () {  
8      addMouseListener (new MouseListener () {  
9        public void mouseClicked (MouseEvent e) {}  
10     });  
11   } // end constructor  
12   
13 } // end class H2ClassJ

In line 8, addMouseListener needs methods implemented since it derived from the interface MouseListener, such as moussedPressed(MouseEvent e), mouseReleased(MouseEvent e), mouseEntered(MouseEvent e), and mouseExited(MouseEvent e). All these methods, including mouseClicked(MouseEvent e) must be overridden for the code to work. Also, the value given for serielVersionUID is not of a long type. The variable type must be changed to either a constant or an int for the code to work.

10. (10 pts) Why does the following code give a compiler warning? (Use javac -Xlint) How should it be fixed?

1 import javax.swing.\*;  
2   
3 public class H2ClassK {  
4   String [] sa = {"a", "b", "c"};  
5   JComboBox jcbA = new JComboBox (sa);  
6 } // end class H2ClassK

*Raw use of parameterized class 'JComboBox'*

*Raw use of parameterized class 'JComboBox'*

*Unchecked call to 'JComboBox(E[])' as a member of raw type 'javax.swing.JComboBox'*

These warnings appear because JComboBox is a generic class and must have a generic type inside.

The following should be done in line 5:

JCombobox<String> jcbA = new JCombobox<>();

**Grading Rubric:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Meets** | **Does not meet** |
| Problem 1 | **10 points** Gives a one or two sentence description of each standard UML diagram. | **0 points** Does not give a one or two sentence description of each standard UML diagram. |
| Problem 2 | **10 points** Explains how the toString methods in the classes H2ClassA and H2ClassB be written to give the indicated output and take advantage of the natural toString method in H2ClassB. | **0 points** Does not explains how the toString methods in the classes H2ClassA and H2ClassB be written to give the indicated output and take advantage of the natural toString method in H2ClassB. |
| Problem 3 | **10 points** Provides at least two good answers explaining how the code can be corrected. | **0 points** Does not provide at least two good answers explaining how the code can be corrected. |
| Problem 4 | **10 points** Explains why the code gives a compiler error.  Explains how the code should be fixed. | **0 points** Does not explain why the code gives a compiler error.  Does not explain how the code should be fixed. |
| Problem 5 | **10 points** Explains what is wrong with the declaration.  Explains how the code should be fixed. | **0 points** Does not explain what is wrong with the declaration.  Does not explain how the code should be fixed. |
| Problem 6 | **10 points** Explains what is wrong with the code.  Explains how the code should be fixed. | **0 points** Does not explain what is wrong with the code.  Does not explain how the code should be fixed. |
| Problem 7 | **10 points** Explains what is wrong with the code.  Explains how the code should be fixed. | **0 points** Does not explain what is wrong with the code.  Does not explain how the code should be fixed. |
| Problem 8 | **10 points** Explains what is wrong with the code.  Explains two ways x could be given a values of 24 legally. | **0 points** Does not explain what is wrong with the code.  Does not explain two ways x could be given a values of 24 legally. |
| Problem 9 | **10 points** Explains what is wrong with the code.  Explains 2 effective ways the code could be fixed. | **0 points** Does not explain what is wrong with the code.  Does not explain 2 effective ways the code could be fixed. |
| Problem 10 | **10 points** Explains why the code gives a compiler warning.  Explains how it should be fixed. | **0 points** Does not explain why the code gives a compiler warning.  Does not explain how it should be fixed. |