UNIVERSITY
OF SKÖVDE

School of Informatics

WRITTEN EXAMINATION

Course Object Oriented Programming G1F

Examination

Course code IT401G Credits for written examination 2,5hp

Date 2024-03-18 Examination time 14:15–18:30

Examination responsible Simon Butler

Teachers concerned András Márki

Aid at the exam/appendices Swedish-English/English-Swedish dictionary

Other

Instructions

- ☐ Take a new sheet of paper for each teacher.
- ☐ Take a new sheet of paper when starting a new question.
- ☒ Write only on one side of the paper.
- ☒ Write your name and personal ID No. on all pages you hand in.
- ☒ Use page numbering.
- ☒ Don´t use a red pen.
- ☒ Mark answered questions with a cross on the cover sheet.

Grade points

Examination results should be made public within 18 working days

Good luck!

UNIVERSITY
OF SKÖVDE

Note: The examination includes true-false questions. You must mark each statement true or false for the true-false questions, which use square icons. Enter your answers to true-false questions directly into the exam paper. Answers given otherwise do not count. Select an option by drawing a cross in the box. If you change your mind, fill in the whole box.

Properly completed true-false responses: 

Amended true-false response: 

Marking:

The exam is divided into two sections. The first section consists of 5 questions and a total of 50 points. The second section consists of longer questions with a combined total of 70 points. (The maximum score for the exam paper is 120 points.) To pass the exam you are required to score 60 points or more.

Total number of pages : 12

Name ..... Personal ID No. .....

Part 1

Question 1 (10 points)

Which of the following statements are true or false about object-oriented programming? Fill in the box to indicate T (true) or F (false).

T	F		
☐	☐	(a)	A class is used as a template to instantiate an object.
☐	☐	(b)	Method overriding depends on static binding.
☐	☐	(c)	Encapsulation means that a class stores its data in public fields.
☐	☐	(d)	All Java classes inherit methods from the `java.Object` class.
☐	☐	(e)	Overriding a method implemented in a superclass will change the behaviour of all instances of that superclass.

Question 2 (10 points)

Which of the following statements are true or false about exceptions and exception handling in Java? Fill in the box to indicate T (true) or F (false).

T	F		
☐	☐	(a)	A **`finally`** block will be executed whether an exception is thrown or not.
☐	☐	(b)	Exceptions that are not caught by a program will cause the program to stop running.
☐	☐	(c)	Tests of the form **`if`** `("hello".equals(testString))` are considered to be bad practice because a `NullPointerException` can be thrown by the test.
☐	☐	(d)	The Java compiler warns the developer when a method can throw an unchecked exception.
☐	☐	(e)	Errors such as `OutOfMemoryError` can be caught and handled by the developer in the same way as exceptions.

Name ..... Personal ID No. .....

Question 3 (10 points)

Which of the following statements about programming paradigms are true or false?
Fill in the box to indicate T (true) or F (false).

T	F		
☐	☐	(a)	The Lambda syntax, or Lambda expressions, are used to implement procedural programming in Java.
☐	☐	(b)	Object-oriented programming offers several advantages over procedural programming, including better organization, more flexibility, and greater modularity.
☐	☐	(c)	SQL queries are an example of logic programming.
☐	☐	(d)	C#, C++ and SmallTalk are object-oriented programming languages.
☐	☐	(e)	Prolog is a functional programming language.

Question 4 (10 points)

Which of the following statements are true or false about Java class, field and method declarations?
Fill in the box to indicate T (true) or F (false).

T	F		
☐	☐	(a)	The Java compiler automatically provides a no-argument, default constructor for any class without constructors.
☐	☐	(b)	The statement: `public class Foo implements Bar` means that `Bar` is the subclass of `Foo`.
☐	☐	(c)	A class with only private constructors can be instantiated from a public static method within the class.
☐	☐	(d)	A class may implement more than one interface, e.g. `class MyClass implements Runnable, ActionListener`.
☐	☐	(e)	A Java method signature consists of the number and type of arguments, and the return type.

Name ..... Personal ID No. .....

Question 5 (10 points)

Which of the following statements are true or false about multi-threaded Java applications?
Fill in the box to indicate T (true) or F (false).

T	F		
☐	☐	(a)	Concurrent execution is possible on all hardware that supports parallel execution.
☐	☐	(b)	Deadlock means two threads can write the same data simultaneously, resulting in data corruption.
☐	☐	(c)	The Java monitor model only works for static functions.
☐	☐	(d)	The `volatile` keyword ensures atomicity.
☐	☐	(e)	In Java, an instance of a class that implements the `Runnable` interface can run as a thread.

Name ..... Personal ID No. .....

Part 2

Question 6: Object Oriented Design (1) (10 points)

Which of the following statements are true or false about Object-Oriented Design? Fill in the box to indicate T (true) or F (false).

T	F		
☐	☐	(a)	Inheritance links in UML class diagrams symbolise a "is-a" relationship.
☐	☐	(b)	Abstraction means that the details of how an object works are available to the user.
☐	☐	(c)	"Loose coupling" allows library implementations to be changed while using the same return type for application programming interface (API) methods.
☐	☐	(d)	`<<extends>>` in a UML use case diagram shows an action is an optional component of the action that extends it.
☐	☐	(e)	Generally, a constructor should call another method to initialise the object.
☐	☐	(f)	Interfaces specify class behaviour and, with exception of default methods, require that programmers implement the behaviour in a concrete class.
☐	☐	(g)	Encapsulation allow the programmer to control how data can be updated and read.
☐	☐	(h)	Java best practice is to override the default constructor to define behaviour and prevent the compiler making assumptions.
☐	☐	(i)	Access modifiers are used to control the visibility of attributes and methods, and classes.
☐	☐	(j)	The use of the Comparator class to sort collections in Java is an example of the Strategy design pattern.

Name ..... Personal ID No. .....

Question 7: Object Oriented Design (2) (10 points)

Which of the following statements about the class diagram are true or false?

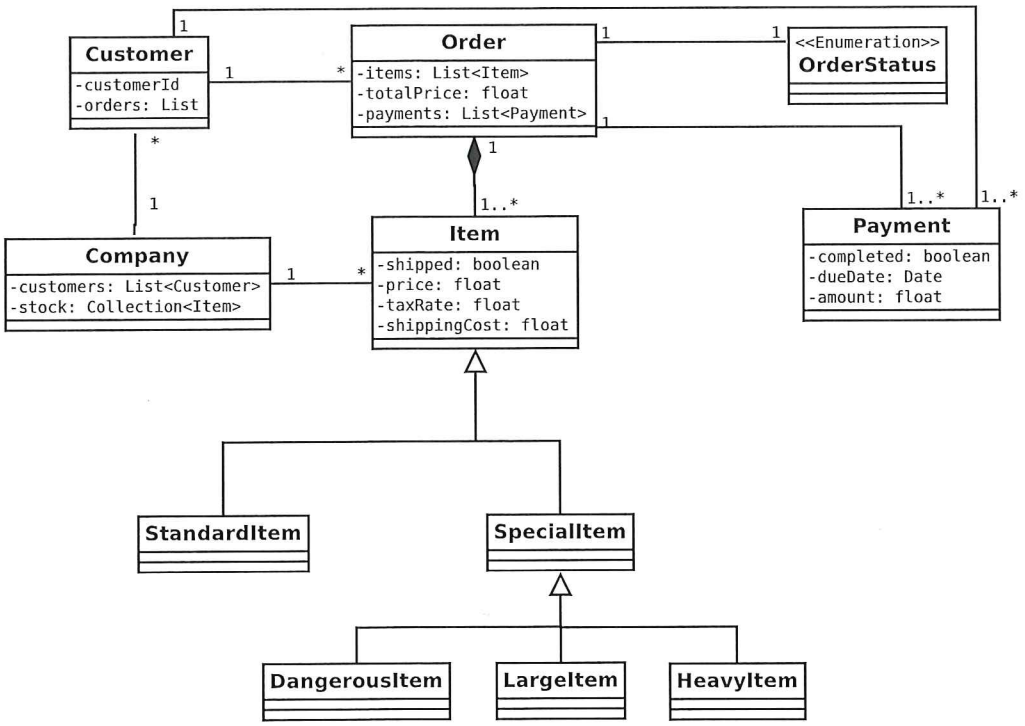

Figure 1: UML diagram

T	F		
☐	☐	(a)	A Customer can make an order that includes StandardItems and SpecialItems.
☐	☐	(b)	Customers can make more than one payment for each order.
☐	☐	(c)	The diagram shows that Customers do not pay tax when buying SpecialItems.
☐	☐	(d)	A Company must have Customers.
☐	☐	(e)	An instance of DangerousItem contains information about which order it is part of.

Name ..... Personal ID No. .....

Question 8: Requirements Analysis (10 points)

A book is written by an author, who may need the services of an illustrator to draw pictures. A book is published by a publisher, sold by a bookshop, and read by a reader. For a reader to read a book, she/he must buy it from a bookshop that sells the book and has a copy of the book in stock.

Draw a use case diagram for this scenario, showing relationships between different use cases.

Name ..... Personal ID No. .....

Question 9: Code Writing and Interpretation (10 points)

Consider the following Java program.

(a) (1 point) The program does not compile and run as written because of one error in the code. How must the code be revised so that it can be compiled and executed? State the line number where your revision should be made.

(b) (4 points) Assuming all errors in the code have been fixed, and the code compiles: draw the GUI window that is created when the application is executed, showing the placement of any buttons, text boxes and other widgets.

(c) (4 points) Explain how to add functionality to raise the first number to the power of the second, e.g. $2^3 = 8$, $3^2 = 9$ and $10^3 = 1000$. State the line numbers where any additional code should be inserted and write the code that needs to be inserted at that point

(d) (1 point) The program contains some examples of bad practice. Identify one example, and explain why it is bad practice. (The starred import on line 1 does not count.)

```
1  import javax.swing.*;
2  import java.awt.event.ActionEvent;
3  import java.awt.event.ActionListener;
4  import java.awt.FlowLayout;
5
6  public class SimpleCalculator extends JFrame {
7      private JTextField number1Field;
8      private JTextField number2Field;
9      private JComboBox<String> operationBox;
10     private JButton calculateButton;
11     private JLabel resultLabel;
12
13     public SimpleCalculator() {
14         super("Simple Calculator");
15         number1Field = new JTextField(5);
16         number2Field = new JTextField(5);
17         String[] operations = {"+", "-", "*", "/"};
18         operationBox = new JComboBox<>(operations);
19         calculateButton = new JButton("Calculate");
20         resultLabel = new JLabel("Result: ");
21
22         setLayout(FlowLayout());
23         add(number1Field);
24         add(operationBox);
25         add(number2Field);
26         add(calculateButton);
27         add(resultLabel);
28
29         calculateButton.addActionListener(new ActionListener() {
30             @Override
31             public void actionPerformed(ActionEvent e) {
32                 try {
33                     double number1 = Double.parseDouble(number1Field.getText());
34                     double number2 = Double.parseDouble(number2Field.getText());
35                     String operation = (String) operationBox.getSelectedItem();
36                     double result = 0;
37
38                     switch (operation) {
39                         case "+":
40                             result = number1 + number2;
41                             break;
42                         case "-":
43                             result = number1 - number2;
44                             break;
45                         case "*":
46                             result = number1 * number2;
47                             break;
```

Code listing continues on next page.

Name ..... Personal ID No. .....

```
48                          case "/":
49                              if (number2 == 0) {
50                                  resultLabel.setText("Error: Divide by Zero");
51                                  return;
52                              }
53                              result = number1 / number2;
54                              break;
55                      }
56                      resultLabel.setText("Result: " + result);
57                  } catch (NumberFormatException ex) {
58                      resultLabel.setText("Error: Invalid Number");
59                  }
60              }
61          });
62
63          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
64          setSize(220, 200);
65          setVisible(true);
66      }
67
68      // Program entry point
69      public static void main(String[] args) {
70          new SimpleCalculator();
71      }
72  }
```

Name ..... Personal ID No. .....

Question 10: Problem solving (20 points)

You should model an office ice cream machine and several programmers in that office. The programmers either code, think or eat ice cream. Programmers can be lactose intolerant; if they eat an ice cream containing lactose, they will get stuck in the thinking state. The length of time spent in each of the three states is chosen randomly and should be displayed for the individual programmer with a simple standard printout (writeout). The office ice cream machine can only have a single instance: If a programmer comes, the ice cream machine creates a new ice cream that can optionally be lactose-free. The machine supports only vanilla flavour for the time being. The programmers can choose their preferred ice cream by selecting if it should be lactose-free. Only one programmer can use the machine at a time, and the machine creates a new, specified ice cream when a programmer comes to the machine and asks for it. (The ice cream machine does not have a buffer with pre-created ice cream.)

(a) (5 points) Write the Java code for the class representing the office.

(b) (5 points) Write the Java code for the class representing the ice cream machine.

(c) (10 points) Write the Java code for the class representing the programmers. (Use the outline of the Programer class below to get started and add any fields and methods required for your solution.)

```
public class Programmer extends Thread {
   // single random number number generator shared by all programmers
   private static Random random = new Random();

   private String currentState;
   private String name;
   private boolean isLactoseIntolerant;
   private IceCreamMachine

   // TODO: implement constructor

   // TODO: implement run() method
   public void run() { ... }

   // TODO: implement any other required methods

}
```

Name ..... Personal ID No. .....

Question 11: Threads (10 points)

Examine the `IntegerHandler` class below.

```
1  import java.util.ArrayList;
2
3  public class IntegerHandler extends Thread {
4      public static ArrayList<Integer> conveyorBelt = new ArrayList<>();
5
6      private int id;
7
8      public IntegerHandler(int id) {
9          this.id = id;
10     }
11
12     public void addNumberToConveyorBelt(int number) {
13         System.out.println("Updating conveyorBelt");
14         conveyorBelt.add(number);
15     }
16
17     public Integer removeAndReturnNumberFromConveyorBelt() {
18         System.out.println("Updating conveyorBelt");
19         if(!conveyorBelt.isEmpty())
20             return conveyorBelt.remove(0);
21         return null;
22     }
23
24     public void run() {
25         for (int i = 0; i < 5; i++) {
26             Integer removedNumber = removeAndReturnNumberFromConveyorBelt();
27             if (removedNumber != null)
28                 System.out.println("ID" + id + ":The number on the conveyor was:"
29                                    + removedNumber);
30             else
31                 System.out.println("The conveyor was empty");
32         }
33     }
34 }
```

(a) (2 points) If a single `IntegerHandler` instance in the system is executed after the `conveyorBelt` initially has the `Integers [4, 3, 2, 1]` stored, what would be the output after executing that single `IntegerHandler` thread, considering that nothing else is modified within the software solution?

(b) (2 points) Why is this solution not thread-safe? Motivate your answer!

(c) (6 points) What and how would you modify to make the solution thread-safe? You only need to include the code you need to modify, as long as you mark clearly what changes should be done and where in the code.