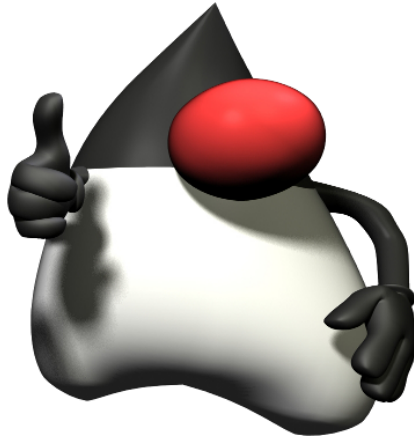


Examiner Tobias Andersson Gidlund	
Date 2022-05-20	
Time 8-13	
Place Various	
Course Code 1DV502	
Allowed aids No aids allowed on campus. Distance students can use books, programs, internet and so on but not ask anyone else to do the exam.	<p>Messages from the teacher</p> <p>This exam has only a few questions. The purpose of the exam is to see that you have understood the principles of object oriented programming and therefore the questions will be of a rather practical nature.</p> <p>Write in a way so that it is easy for us to read! This means that if you do it on paper, and there are lines on the paper, you let your letters sit on those lines. Take more paper! Only answer <i>one</i> question per sheet of paper – it is okay to answer a, b and c (and so on) on the same sheet of paper, but do not squeeze everything in – pick another sheet of paper instead.</p> <p>Use the time provided! Make a new drawing or code on a new sheet of paper if you have the time. Do not hand in papers with loads of ink “corrections” if you have time to re-write it in a cleaner way.</p> <p>You will not need to write code per se, but you might, from time to time, need to provide “Java like” code to show us that you have understood the ideas. With this, we mean code that is similar in structure to Java, but we will not mind if a semicolon is wrong, or if the order of parameters to built-in method is wrong.</p> <p>You can answer in Swedish or English.</p> <p>Make assumptions where you believe the information provided by us is not enough. Also explain when you do, so that we, when correcting, can follow your line of thought.</p> <p>Campus Pass is 24 points (E – C grade) Pass with distinction is 32 points (A or B grade).</p> <p>Distance Pass is 32 points (E – C grade) Pass with distinction is 36 points (A or B grade).</p>
Messages from the teacher This exam has only a few questions. The purpose of the exam is to see that you have understood the principles of object oriented programming and therefore the questions will be of a rather practical nature.	
Write in a way so that it is easy for us to read! This means that if you do it on paper, and there are lines on the paper, you let your letters sit on those lines. Take more paper! Only answer <i>one</i> question per sheet of paper – it is okay to answer a, b and c (and so on) on the same sheet of paper, but do not squeeze everything in – pick another sheet of paper instead.	
Use the time provided! Make a new drawing or code on a new sheet of paper if you have the time. Do not hand in papers with loads of ink “corrections” if you have time to re-write it in a cleaner way.	
You will not need to write code per se, but you might, from time to time, need to provide “Java like” code to show us that you have understood the ideas. With this, we mean code that is similar in structure to Java, but we will not mind if a semicolon is wrong, or if the order of parameters to built-in method is wrong.	
You can answer in Swedish or English.	
Make assumptions where you believe the information provided by us is not enough. Also explain when you do, so that we, when correcting, can follow your line of thought.	
Campus Pass is 24 points (E – C grade) Pass with distinction is 32 points (A or B grade).	
Distance Pass is 32 points (E – C grade) Pass with distinction is 36 points (A or B grade).	

Exam in Object Oriented Programming, 1DV502, 7,5hp

May20, 2022, 8.00–13.00

1. Object Oriented Modelling

You have been tasked with the mission to create a system for a car rental company. The company has a fleet of twelve cars, divided into five so called “small” cars (like Ford Fiesta), four “medium” sedan cars (like Volvo S60) and three “large” cars with space for a lot of luggage (like Audi A4 Avant). They will later, hopefully, extend the car fleet with at first two Tesla Model S and at the end of the year two more.

All of these cars will be rented to customers where it is important that the company knows which customer has what car rented for what time. This is important because the car renting company wants to be able to rent out a car again as soon as possible. One customer can rent several cars at the same time and a part from keeping track of what cars are out right now, the system also needs to keep a history of what an individual customer has rented.

The system needs to separate private and enterprise customers, that is if the car is rented by a private individual or by a company. Company customers can get a reduced price after renting for a long time (more than ten times). Private customers do not get this, but the company want to be able to send out mail with offers for, for example, the summer season or during holidays.

Feel free to add additional knowledge and assumptions to your understanding and explain them when answering the question. You do not need to provide all getters and setters for all attributes.

- (a) Create a *class diagram* with attributes, operations and relationships. (5p)
- (b) Explain and motivate how and why you are using *encapsulation* in this case. Motivation is important and that will require you to also explain how encapsulation works in the general case. (5p)
- (c) Explain and motivate the relationships in your diagram. Go through each of the relationships and motivate why there is a relationship between the two classes and explain the multiplicity on both sides. (5p)
- (d) Create an *object diagram* of an instance of the system. (5p)



Figure 1: Class diagram for tool for linking buyers and products.

2. The simplified class diagram of a part of a much larger application that is shown in Figure 1 should be implemented using *Java like code*. It shows a **Cart** which has a dependency on **Product**. For this to work, you will need to add additional method(s) and/or attributes to the **Cart** when compared to what is shown in the diagram. Notice that **LocalDate** is a built-in class in Java so you do not need to model this class yourself.
You do not need to provide a **main** method and you *do not* need to provide full encapsulation.
 - (a) Provide an implementation of the diagram above in *Java like code*. You must add a number attributes and methods for it to work, but you do not need to add any additional classes. (10p)
3. The *Iterator pattern* is a behavioural pattern that allows for traversing a number of objects.
 - (a) Explain a bit more on what the iterator pattern is and how the iterator pattern could be used in the application described in task one, if at all. You do not *need* to show a class diagram, but it will make it easier to understand. Use technical terms when discussing and motivating your answer. (5p)
 - (b) Discuss whether using a pattern like iterator could be a good idea for the task one, or why it perhaps isn't. Motivate your answer and use technical details to some extent (class diagram not needed but could be beneficial). (5p)

Good luck!