Kristianstad University
SE-291 88 Kristianstad
Sweden
+46 44 250 30 00
www.hkr.se

Faculty of Natural Science
Andreas Nilsson

Fundamental Programming (DA110D)
Home exam (2018-12-21)

Important

This document contains the tasks to be solved during the home exam but before starting to solve the tasks the examinee shall read the information supplied on Canvas.

- Allowed aids, rules, requirements and grading
- Upload information

Task 1

Instead of spending money at cinema tickets this holiday you have decided to invite all your friends to your personal cinema. You have hastily decorated the sitting room with a lot of chairs and now you only need a booking system so that your friends can book a chair for the opening night.

You have placed the chairs in three rows with five chairs in each row (see Figure 1).

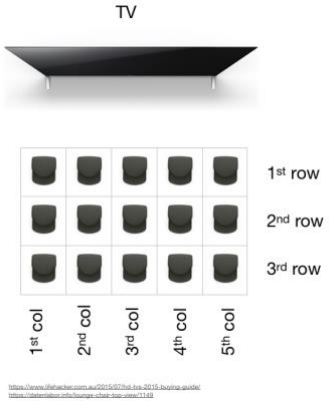

Figure 1 - Cinema layout

Create the program according to the following requirements:

1) Use a multidimensional array to represent the sitting room. Make sure the array is 3x5 (*rows* x *cols*) and that it can store boolean values.
2) Create a menu that has two options. The first option is to book a seat and the second option is to view the currently booked seats.
3) If the user choses to book a seat, a message shall be printed to the screen asking the user to enter the row and column to book. The row shall be read as a number from 1 to 3 and the column shall be read as a number from 1 to 5. Write a message to the screen telling the user if the seat has already been booked by someone else or if the booking was successful. Do not forget to mark the seat as booked.
4) If the user choses to view the bookings a graphical representation of the sitting room shall be printed to the screen. The booked seats shall be represented by an O and all other seats with a . (dot). The output shall also contain a tag <TV> to indicate where the TV is placed. See the video for an example.
5) After having booked a seat or viewed the current bookings the user shall be returned to the menu so that the user can make another choice.
6) The program shall not have an exit option (infinite loop).

The program must use at least:

- One class (the main class)
- A multidimensional array
- A loop
- An if-statement

Video example of running the finished program.

Task 2

Around Christmas time people often send Christmas cards to their friends and family. You have decided to create a program that can keep track of the cards you receive this year. You want to keep track of the different types of cards you receive. Even if you do not know all the different types of cards you will receive, you will start by adding one type.

Create the program according to the following requirements:

1) Create a class names Card that has only one instance variable, a String that stores the senders name. The name shall be sent as a parameter to the constructor and it shall be accessible via a getter method.
2) Create a class to represent hand drawn cards, call it HandDrawn. The class shall inherit from the Card class and shall have one more instance variable called *niceDrawing*. The instance variable shall be a boolean, shall be sent as a parameter to the constructor and it shall also be possible to both get the value and to set a new value.
3) Create an ArrayList of cards where you can store any card type. This means that the list shall not only store cards from the HandDrawn class but also any other cards that are subclasses to the Card class.
4) Create two HandDrawn objects using the data from Table 1.

Table 1 - Card data

senderName	niceDrawing
Anna	True
Kalle	False

5) Add the two objects to the ArrayList.
6) Create a method that takes the list of cards as a parameter and prints all the cards to the screen. Each card shall print all stored information.
7) Use the newly created method to print all the cards. Use the ArrayList of Card objects as a parameter.

The program must use at least:

- Three classes (main + two others)
- An ArrayList
- Inheritance
- Two objects
- A method in the main class taking an ArrayList as parameter

Video example of running the finished program.

Task 3

Christmas presents needs a label attached to them so that Santa knows who the present is intended for. You have come up with the clever idea to create a program that can generate the labels for you.

Create the program according to the following requirements:

1) The program shall start by asking the user to input the sender's initials as a String. Use the keyboard to read the input.
2) Next, the program shall ask the user to input the recipient's initials using the keyboard.
3) Create a method taking the two Strings as parameters. The method shall return a String. The returned string shall be the Christmas present label (as a String). The label shall be a multiline String in the following format.

```
**************
* From: A.B. *
* To:   C.D. *
**************
```

   The initials shall always be entered using the format highlighted in red above. The examinee does not have to check the format but can expect the initials to always use the correct format (to simplify the creation of the label).
4) Call the newly created method from the main method, use the two read Strings with the initials as parameters. Store the returned String in a local variable.
5) Print the returned Christmas present label to the screen in the main method.

The program must use at least:

- One class (the main class)
- One method, taking two parameters and returning something

Video example of running the finished program.

Task 4

The local politicians have decided to give away free Christmas food to everyone this year. They have tasked you to create a program that can be used to keep track of people in the queue. The politicians are sure no more than 100 persons will ever be in the queue at any time.

Create a program according to the following requirements:

1) The queue shall be represented using an array of Strings. The array shall be able to store 100 elements and each element shall store the name of a visitor in the queue.
2) Create a menu where the user can register a new visitor (place them in the queue) or to advance the queue (remove a visitor from the queue).
3) A loop shall be placed around the menu so that the user can continue using the menu over and over again.
4) If the option to register a new visitor is chosen, the program shall ask the user to input the name of the visitor and then save the name in the array. The name shall be stored in the first index not already containing a name.
5) Should the user select the option to advance the queue a message shall be printed to the screen informing everyone who is now served. It is always the visitor at index zero who is the next to be served. The message could look similar to this message:

   ```
   Now serving: Anna
   ```

6) When advancing the queue, not only should the next visitor's name be displayed but the actual content in the array should also update. The visitor at index zero should be removed, all other visitors in the queue should take one step forward. I.e. the visitor previously stored at index one now moves to index zero, the user previously stored at index two now moves to index one and so on. See Table 2 for an illustration.

   Table 2 - Queue before and after serving Anna

Index	Before serving Anna	After serving Anna
[0]	Anna	Kalle
[1]	Kalle	Pelle
[2]	Pelle	

7) If the user tries to advance the queue when the queue is empty the following message shall be printed to the screen.

   ```
   No visitors in queue
   ```

The program must use at least:

- One classes (the main class)
- An array
- A loop

Video example of running the finished program.