KIT101 Programming Fundamentals

PP Task 7.2 Structure Charts

Overview

Purpose:	Learn how create a structure chart to document the methods within a program.
Task:	Create a structure chart for your program from PP Task 7.1.
Learning Outcomes:	2 4 5
Time:	Aim to complete this task before the start of Week 8.
Resources:	• Introductory Programming Notes: ◦ 13 Functional Decomposition

Note: This task should be attempted after you are satisfied with your solution to PP Task 7.1. You do not need to wait until you have had 7.1PP formally assessed.

Submission Details

Upload the following to the MyLO submission folder for this task:

- An **image** of your structure chart (photo or scan)

Assessment Criteria

A Completed submission will:

- Show all methods declared within your main program for 7.1PP (and *may* show methods in your data class if they are called explicitly)
- Not show methods declared outside your source code
- Have arrows indicating calls between methods
- Have data shown next to the arrows indicating the direction the data is travelling

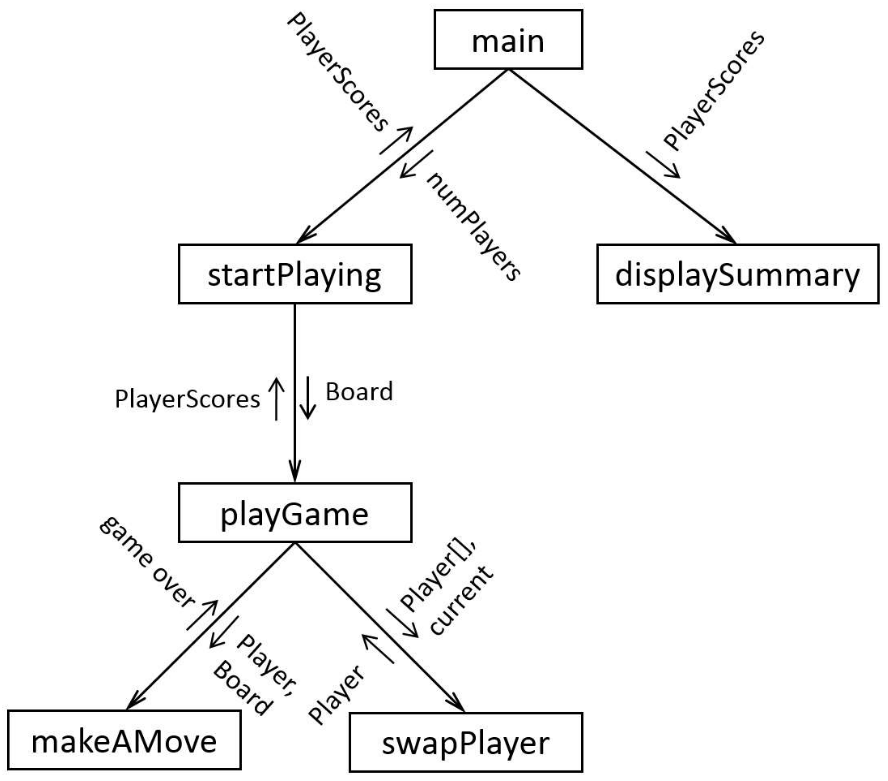
Instructions

Create a structure chart for the program you created in PP Task 7.1.

Structure charts provide an effective way of communicating the methods within a program. To create a structure chart you would do the following (an example is shown below, and more details are given in Part 13 of the *Introductory Program Notes* on MyLO):

1. Draw a box for the program's main() method at the top of the page
   - Write the text *main* within the box; this now represents the main method
2. Draw a box for each of the methods that main() calls. Position these below the box for *main* so that you can easily draw arrows from the bottom of main to the methods it calls
   - Write the name of each method within the box
   - Draw an arrow from main to the method's box
   - Next to the line draw smaller arrows to indicate the data that will be passed to, or returned from, the method

Here is an example for a fictitious program (meaning you will not have seen source code for this), where you may assume values with an initially upper case name represent a custom data type:



This kind of diagram can be used to either communicate the structure of an existing program or as a design tool to help you think about how your program will be organised.

Note: No need to include nextInt() or next() for example, as those are in the Scanner class. Do include readReading (readExpense, etc.) as it is in the program’s code.