# Data Structures and Java

## Assignment 3 – Classes and Objects

### Assignment objectives:

In this assignment, you will use your knowledge of the following new concepts:

1. Create class templates
2. Create object instances of user written classes
3. Use multiple .java files in a project to demonstrate object to object interactions
4. Use a flexible ArrayList to store object instances

You may also need the following old concepts:

1. Reading from console (you can use the Scanner)
2. Loops (while) and condition flows (if-else if-else)
3. Calling methods and sending parameters

### Requirements

In this assignment you will create a simple NHL game management system which allows users to manage players on the home team’s roster for each game. We will create the following classes: GameManagementSystem, Game and Player.

**Important:** You do not have to worry about using try-catch-finally blocks for user input. Rather, assume users enter valid input.

Create a project called A3FirstLastName.

#### Requirement 1- class GameManagementSystem

Create a class called GameManagementSystem.

The class, GameManagementSystem, will contain the main() method. It should print the following menu:

*Please select from the following:*

1. *Add a game*
2. *Add a player to the home team roster for a game*
3. *Add a player to the visiting team roster for a game (not implemented)*
4. *View available games*
5. *Exit*

**If the user selects option 1**, ask the user to enter information to be stored in the following variables: int gameNumber, String homeTeamCode and String visitingTeamCode.

E.g.:

*Please enter the game number:*

*20026*

*Please enter the home team three letter code:*

*CHI*

*Please enter the visiting team three letter code:*

*T.B*

These inputs will be sent to a constructor in the Game class that accepts three inputs, and creates a new Game object (store this in a variable called Game currGame).

Use an instance of the ArrayList class called gameArr to store the newly added game. To create an ArrayList object:

ArrayList<Game> gameArr = new ArrayList<>();

To add the game object:

gameArr.add(currGame);

**If the user selects option 2** and there are no games added to the system, print the message: *"There are no games in the system."* To check this, use the .isEmpty() method on the ArrayList.

If there are games, ask user to enter the game number for which to add a player. Store this in the gameNumber variable. We have to search through our gameArr ArrayList to see if the given game number exists. To do this, we have to extract each Game object one by one – we can use the method called .get() in the ArrayList class that takes one input which is the index location as an int and returns the object at that location. Once you have a Game object, we can invoke its methods using the dot operator – here, we want to call the getGameNumber() method in the Game class which returns the game number as an int. We can then check to see if the game number matches the game number that the user wants to add a player to.

We can additionally use a variable boolean isGameFound to keep track of whether we find the game while searching for it. Once a game is found in the list of games, there is no need to continue searching in the ArrayList.

When the game is found, check to see if there are open positions on the home team roster (i.e. there is a maximum of 18 players allowed on a roster for a game). If there are positions available, then continue to obtain player details from the user. Ask the user to enter the player information and store it in two (2) variables: String playerName, char playerPosition. Note that there are 5 valid playerPosition values, ‘G’ (goaltender), ‘C’ (center), ‘R’ (right wing), ‘L’ (left wing) and ‘D’ (defense).

Concatenate this into a single variable String newPlayer with tabs in between the values.

Below is some sample code to perform the above steps:

Game currGame;

for (int i = 0; i < gameArr.size() && isGameFound == false; i++) {

// below you will write code to search through gameArr

// to see if desired game exists

// set the value of currGame to equal the object retrieved from index i of gameArr

if (currGame.getGameNumber() == gameNumber) {

// we found a matching game

// if home roster is not full

//ask for player information and add to game

isGameFound = true;

}

}

To add the player to the game, call the addPlayer () method in Game and send it the String newPlayer. The method addPlayer() should return the updated total number of players on the home team roster as an int. You can store it in int homeTeamRosterSize and print the following to the user if a player is successfully added (that is, if the roster is not already full. Otherwise, print an error message).

*"Player successfully added. The game " + gameNumber + " now has " + homeTeamRosterSize + " players on the home roster"*

Or

*“Sorry, the roster is full. Cannot add player.”*

After searching through gameArr, if the game requested by user is not found, print a message: *“Sorry, that game was not found. Please try again."*

**If the user selects option 3,** output a message that this option has not been implemented: *“Sorry, this option is not yet implemented. Please try another option.”*

**If the user selects option 4**, check .isEmpty() for the gameArr ArrayList object, and if so print the message: *“No games to list."* If there are games, then use a for loop to go through gameArr, retrieve each game object using the get() method and invoke its three getter methods - getGameNumber(), getGameDescription() and getRosterSize() to obtain game information and to print it. Also print the current roster size as shown in the sample code below.

for (int i = 0; i < gameArr.size(); i++) {

// use the getters from Game to get game information

}

#### Requirement 2 – class Game

This class has no main() method.

It contains several private fields:

private final int gNumber;

private final String hTeamCode;

private final String vTeamCode;

private final int rosterMaxSize = 18;

private final ArrayList<String> homeRoster = new ArrayList<String>();

It contains one constructor:

public Game (int gameNumber, String homeTeamCode, String visitingTeamCode){

// here you will set the value of the private fields

}

It contains the following public methods (you have to write code to complete them):

public int getGameNumber () {} //returns private field gameNumber

public String getGameDescription () {} //returns vTeamCode + “ vs. “ + hTeamCode

public int getRosterMaxSize() {} //returns private field rosterMaxSize

public int getRosterSize(boolean isHome) {} //returns the current size of the home/visitor roster

public int addPlayer(String player, boolean isHome) {} // adds player to homeRoster/visitorRoster and returns the size of the roster as an int

You may add other methods and fields as needed.

#### Requirement 3 – Add feature to print home roster

Write code to include one more option on the menu in GameManagementSystem to view the home roster for a given game number. The logic is as follows:

Check to see if the game exists. If not, print an error message.

If it exists, then we want to retrieve the private ArrayList (homeRoster) that contains the objects that contain the player information in Game.

If the homeRoster is empty, print a message.

If not empty, then retrieve the player information from the objects that contain the player information.

#### Requirement 4 – Refactor String representation of player into a specific class, Player

This class should contain private fields - String playerName, char playerPosition. Thus, rather than treat player (information) as a String, you will be treating player as an object with its own class and fields.

Create the necessary getter and setter methods.

Now, when you have to add a new player to a game, create a new Player object, and add it to the arrayList homeRoster. Additionally, now homeRoster is an ArrayList of Player objects, rather than one of String objects.

private ArrayList<Player> homeRoster = new ArrayList<>();

### Deliverables

Please zip your entire project folder containing the **src** sub-folder and the **nbproject** subfolder (the others are not necessary). Name the zip file **A3FirstNameLastName.zip** and submit to Canvas.

### Sample Output

Please select from the following:

1. Add a game

2. Add a player to the home team roster for a game

3. Add a player to the visiting team roster for a game (not implemented)

4. View available games

5. Exit

6. Print home roster

4

No games to list.

Please select from the following:

1. Add a game

2. Add a player to the home team roster for a game

3. Add a player to the visiting team roster for a game (not implemented)

4. View available games

5. Exit

6. Print home roster

2

There are no games in the system.

Please select from the following:

1. Add a game

2. Add a player to the home team roster for a game

3. Add a player to the visiting team roster for a game (not implemented)

4. View available games

5. Exit

6. Print home roster

1

Please enter the game number:

20026

Please enter the home team three letter code:

CHI

Please enter the visiting team three letter code:

T.B

Please select from the following:

1. Add a game

2. Add a player to the home team roster for a game

3. Add a player to the visiting team roster for a game (not implemented)

4. View available games

5. Exit

6. Print home roster

1

Please enter the game number:

20027

Please enter the home team three letter code:

STL

Please enter the visiting team three letter code:

FLA

Please select from the following:

1. Add a game

2. Add a player to the home team roster for a game

3. Add a player to the visiting team roster for a game (not implemented)

4. View available games

5. Exit

6. Print home roster

1

Please enter the game number:

20028

Please enter the home team three letter code:

DAL

Please enter the visiting team three letter code:

WSH

Please select from the following:

1. Add a game

2. Add a player to the home team roster for a game

3. Add a player to the visiting team roster for a game (not implemented)

4. View available games

5. Exit

6. Print home roster

4

Game number: 20026; Description: T.B vs. CHI; Home roster size: 0; Visitor roster size: (not implemented)

Game number: 20027; Description: FLA vs. STL; Home roster size: 0; Visitor roster size: (not implemented)

Game number: 20028; Description: WSH vs. DAL; Home roster size: 0; Visitor roster size: (not implemented)

Please select from the following:

1. Add a game

2. Add a player to the home team roster for a game

3. Add a player to the visiting team roster for a game (not implemented)

4. View available games

5. Exit

6. Print home roster

2

Please enter the game number:

20026

Please enter the player name:

Patrick Kane

Please enter the player position:

R

Player successfully added. The game 20026 now has 1 players on the home roster.

Please select from the following:

1. Add a game

2. Add a player to the home team roster for a game

3. Add a player to the visiting team roster for a game (not implemented)

4. View available games

5. Exit

6. Print home roster

2

Please enter the game number:

20026

Please enter the player name:

Duncan Keith

Please enter the player position:

D

Player successfully added. The game 20026 now has 2 players on the home roster.

Please select from the following:

1. Add a game

2. Add a player to the home team roster for a game

3. Add a player to the visiting team roster for a game (not implemented)

4. View available games

5. Exit

6. Print home roster

4

Game number: 20026; Description: T.B vs. CHI; Home roster size: 2; Visitor roster size: (not implemented)

Game number: 20027; Description: FLA vs. STL; Home roster size: 0; Visitor roster size: (not implemented)

Game number: 20028; Description: WSH vs. DAL; Home roster size: 0; Visitor roster size: (not implemented)

Please select from the following:

1. Add a game

2. Add a player to the home team roster for a game

3. Add a player to the visiting team roster for a game (not implemented)

4. View available games

5. Exit

6. Print home roster

6

Please enter the game number:

20026

Player name: Patrick Kane; Position: R

Player name: Duncan Keith; Position: D

Please select from the following:

1. Add a game

2. Add a player to the home team roster for a game

3. Add a player to the visiting team roster for a game (not implemented)

4. View available games

5. Exit

6. Print home roster

5

Bye