Thomas Coupe, 201241037, Comp 122

Assignment 1 – Object orientated programming

Part 1 – Dice Game

Summary of the requirements

The dice values and number of points should be displayed per round. The program must also output the winner of each round and also declare any ties or draws across all rounds. After all the rounds have been displayed I must calculate and display the total number of rounds won for each player and also the total points for each player. I also need to calculate the average points for each player at the end of the game and I need to display the player with the highest point's total.

Analysis and design

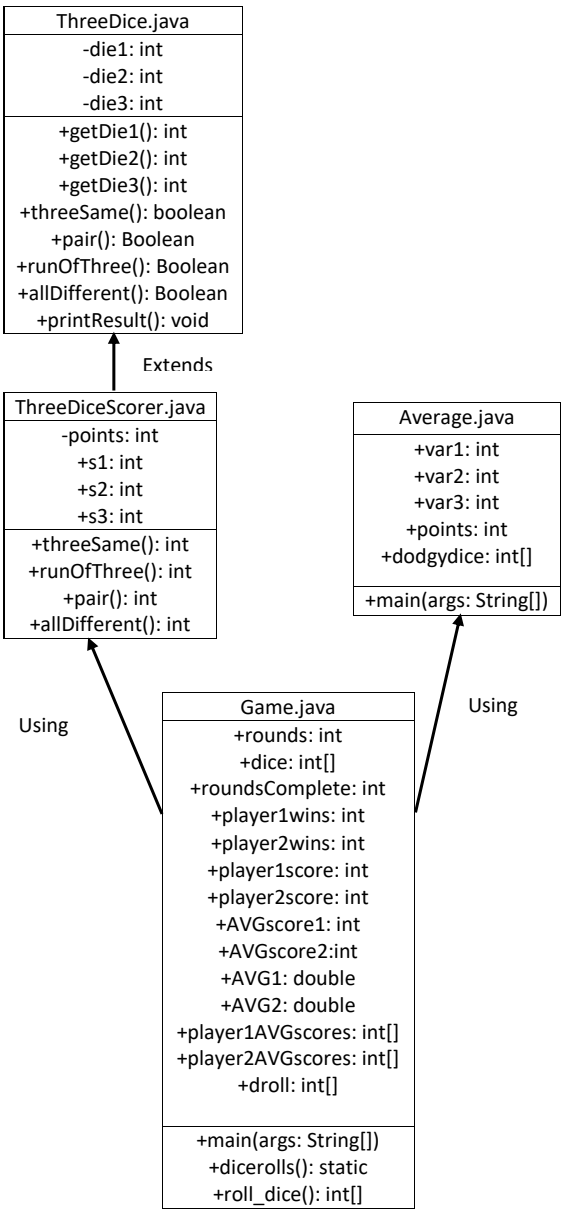
For part one of the assignment I was given a set of requirements in order to make a two player dice game. Firstly I had to make a file called ThreeDiceScorer.java that extended the ThreeDice.java that was given to us, the ThreeDiceScorer.java took the combinations of dice outcomes from ThreeDice.java and gave a points values to each of the outcomes. For example, if player one rolled 3 of the same they would receive an extra 60 points along with the values of the dice being added together. I then moved onto the Game.java which took an input of how many rounds should be played and then played the dice game dependent upon the rounds chosen. For example, if the rounds chosen was "3" then the dice game would be played 3 times. In Game.java I had to use an array for the dice which then used a math.random() to generate each random dice roll. I then had to make another program called Average.java which took the average of all values rolled on a fair die over the whole program and output them to the player at the end of the game. Another requirement was to consider a scenario where one of the dice was unfair and I had to calculate the average. The scenario was to remove a one from the dice and add another 6. To do this I needed to reuse the code from my Average.java however I needed to slightly adjust it in order to account for the unfair die. The table below displays the points for each die roll.

Description	Points calculation	Example	points
Three of the same	Die1 + Die2 + Die3 + 60	2,2,2	6 + 60 = 66
Run of three	Die1 + Die2 + Die3 + 40	3,4,5	12 + 40 = 52
pair	Die1 + Die2 + Die3 + 20	1,4,4	9 + 20 = 29
All different	Die1 + Die2 + Die3	1,5,2	8

T.Coupe@student.liverpool.ac.uk

Thomas Coupe, 201241037, Comp 122

Class Diagrams – Dice Game



T.Coupe@student.liverpool.ac.uk

Thomas Coupe, 201241037, Comp 122

Testing – Dice Game

Test number	Test	Test input	Expected output	Actual results
1	Testing user input	Entering a negative number for the rounds input	The program should not accept the user input and should ask the user to re-input the number of rounds	How many rounds would you like to play? -1 You cannot have a number of rounds less than 0.
2	Testing user input	Entering zero as the number of rounds to be played	The program should actually accept the user input, however as the input was zero the game should stop and close as no rounds are to be played	How many rounds would you like to play? 0 Goodbye, you chose 0 rounds!
3	Testing if the rounds are updated every round played	Entering a positive integer as the rounds to be played and checking that the current round is updated each round	The program should update the round every time a new round is played.	The current round increased by one for each round chosen. This was then displayed to the user to tell them the current round.
4	Testing if points are displayed for each player per round	Using a test number of 5 rounds to see of the players points are displayed for each round.	The program should display both players scores for each round below the current dice rolls.	The points for each player was displayed correctly next to the current die rolls.
5	Testing if total rounds won by each player is displayed at the end of the game	Using the test number 5 to see if the rounds won by each player are displayed at the end of the game.	The program should display the rounds won by player 1 and the rounds won by player 2.	Player 1 won a total of 4 rounds and Player 2 won a total of 1 round. The program displayed the total rounds won correctly.
6	Testing if the average of all dice rolls over the game is calculated and output to both players.	Again I will be using a test input of 5 rounds and I will test to see of the average is correctly calculated for each player at the end of the game.	The program should calculate the average for both players.	Player 1 had an average score of 26.0 Player 2 had an average score of 13.0 The program worked correctly and output the average for both players
7	Testing to see if the player with the most rounds won is declared as the	I will be using a test input of 5 rounds to be played to see if the program successfully	The program should see who has the most rounds won and then declare a	Player 1 won 4 rounds. Player2 won 1 round. Player 1 is the winner!

T.Coupe@student.liverpool.ac.uk

	winner at the end of the game.	outputs the winner at the end of the game.	winner at the end of the game.	The program worked correctly and successfully output the winner at the end of the game.
8	Testing to see if the program displays each winner per round	I shall be using a test input of 5 rounds again to see if the winner of each round is updated.	The program should output the winner of each round.	The program successfully output the winner of each round. No changes to the program needed.
9	Testing if the program can calculate the average score of all possible dice rolls	I shall be running my average.java to see if it can correctly calculate the average.	The program should correctly calculate the average of all possible dice roll scores.	Average: 24.944444444444443 The average output correctly.
10	Testing the average of all possible die scores with an unfair die	I shall be running my average.java to see if it can correctly calculate the average with an unfair die.	The program should correctly calculate the average of all possible dice rolls with one unfair die	Unfair die: 2,3,4,5,6,6 Average: 25.77777777777778 The average was correct, it was slightly higher as there was an extra 6 on the unfair die and it was missing a 1.

Pseudocode – Dice Game

Game.java

```
PUBLIC CLASS Game
    PUBLIC STATIC INT rounds;
    FUNCT MAIN()
    rounds = INT_INPUT
    WHILE true
       IF rounds < 0
           OUTPUT (“You cannot have a number less than zero”)
           EXIT
       IF rounds >= 1
           OUTPUT (“rounds chosen: “+rounds+” Chosen”)
       ELSE IF rounds == 0
           OUTPUT (“Goodbye, you chose zero rounds! “)
    dicerolls()

PUBLIC STATIC VOID dicerolls()
    INT[] dice = new INT[6]
    INT roundscomplete = 0;
    INT player1wins = 0;
    INT player2wins = 0;
    ThreeDiceScorer[] player1r = new ThreeDiceScorer[rounds];
    ThreeDiceScorer[] player2r = new ThreeDiceScorer[rounds];
    INT player1score;
    INT player2score;
    INT AVGscore1 = 0;
    INT AVGscore2 = 0;
    DOUBLE AVG1 = 0;
    DOUBLE AVG2 = 0;
    INT[] player1AVGscores = new INT[rounds]
    INT[] player2AVGscores = new INT[rounds]
    WHILE roundscomplete < rounds
```

T.Coupe@student.liverpool.ac.uk

```
        Dice = roll_dice()
        Player1r[roundscomplete] = new ThreeDiceScorer(dice[0],dice[1],dice[2])
        Player2r[roundscomplete] = new ThreeDiceScorer(dice[3],dice[4].dice[5])
        Player1score = player1r[roundscomplete].getpoints()
        Player2score = player2r[roundscomplete].getpoints()
        Roundscomplete++
        OUTPUT ("round: "roundscomplete". Player1: "dice[0],dice[1],dice[2]" Player2:
"dice[3],dice[4],dice[5]" ")
        OUTPUT ("Player1 score: "player1score "  Player2 score: "player2score)
        AVGscore1 = AVGscore1 + player1score
        AVG1 = AVGscore1 / rounds
        AVGscore2 = AVGscore2 + player2score
        AVG2 = AVGscore1 / rounds
        IF player1score > player2score
            Player1wins++
            OUTPUT ("Player1 won the round")
        ELSE IF player1score < player2score
            Player2wins++
            OUTPUT ("Player2 won the round")
        ELSE
            OUTPUT ("Draw")
    OUTPUT ("Player1 rounds won: "player1wins"   Player2 rounds won: "player2wins)
    IF player1wins > player2wins
        OUTPUT ("Player1 Won the game! ")
    ELSE IF player1wins < player2wins
        OUTPUT ("player2 Win the game! ")
    ELSE
        OUTPUT ("Both players tied, it's a draw! ")
    OUTPUT ("Average score for player1: "AVG1)
    OUTPUT ("Average score for player2: "AVG2)

PUBLIC STATIC INT[] roll_dice()
    INT[] droll = new INT[6]
    FOR (INT I=0; I<6;I++)
        droll[I] = 1 + (INT)(6*Math.random());
    return droll;
```

Commented [TC1]:

Average.java

```
PUBLIC CLASS Average
    FUNCT MAIN()
        DOUBLE points = 0;
        FOR (INT var1=1; var1<=6; var1++)
            FOR (INT var2=1; var2<=6; var2++)
                FOR (INT var3=1; var3<=6; var3++)
                Points += new ThreeDiceScorer(var1,var2,var3).getpoints()

        OUTPUT (points/216)
        points = 0;

        INT[] dodgydice = {2,3,4,5,6,6};
        FOR(INT var1=1; var1<=6; var1++)
            FOR (INT var2=1; var2<=6; var2++)
                FOR (INT var3=0; var3<=5; var3++)
                Points += new ThreeDiceScorer(var1,var2,dodgydice[var3]).getpoints()
        OUTPUT(points/216)
```

T.Coupe@student.liverpool.ac.uk

Thomas Coupe, 201241037, Comp 122

ThreeDiceScorer.java

```
PUBLIC CLASS ThreeDiceScorer EXTENDS ThreeDice
    INT points;
    PUBLIC ThreeDiceScorer(INT s1, INT s2, INT s3)
        Super(s1,s2,s3)
        Points = s1+s2+s3

        IF ThreeSame()
            points = points + 60
        IF runOfThree()
            points = points + 40
        IF pair()
            points = points + 20
        IF allDifferent()
            points = points

    PUBLIC INT getpoints()
        RETURN points
```

Part 2 – Package comparison program

Summary of the requirements

A user should be able to input the number of daytime phone minuets that they use, the number of evening phone minuets that they use and also the amount of broadband usage for a given month. The input must be an integer value and negative values must be disallowed. The program should print the account information for each of the three packages (Bronze, Silver and Gold). The total cost of each type of call, extra broadband usage and also the total cost of that account must the calculated and printed out to the user. The program must recommend the best package for the customer based upon the input.

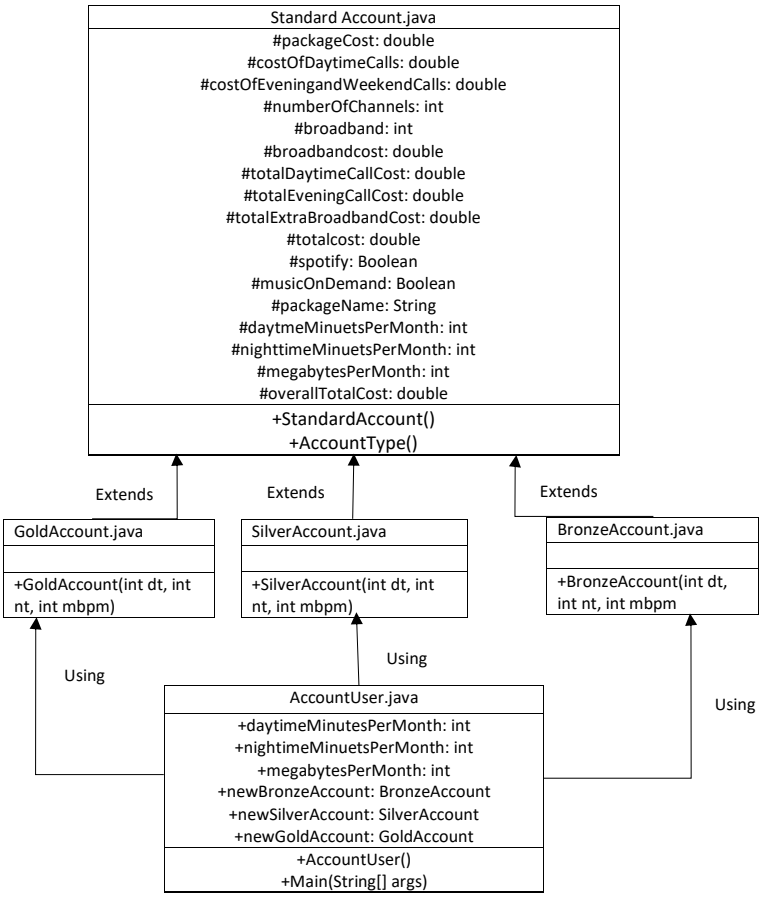
Analysis and design

For this program I am going to have to focus upon the hierarchy of objects within java and I'm going to have to use this hierarchy system in order to make my program. I will first start by defining the values of variables in each of the three accounts. I shall be extended each of the three accounts to another class called the "StandardAccount.java". I shall be using this class to output to the user the information about each package. I also need another class that will take the user input and will output which package is the cheapest to the user, this class will be called "AccountUser.java". overall the program should take the user input and then based upon that input, recommend them the best package.

T.Coupe@student.liverpool.ac.uk

Thomas Coupe, 201241037, Comp 122

Class Diagram



Testing

Test number	Test	Test input	Expected outcome	Actual Outcome
1	Inputting a negative number in the user input Question 1	How many daytime minuets do you use per month? -4	The program should disallow any negative integer inputs	The program disallowed the negative integer and asked the user to re-input a number
2	Inputting a character in the user input Question 1	How many daytime minuets do you use per month? z	The program should crash as I have not declared what happens if a character is input instead of an integer	The program crashed as expected.

T.Coupe@student.liverpool.ac.uk

3	Inputting a negative number in the user input Question 2	How many night time minuets do you use per month? -3	The program should disallow any negative integer inputs	The program disallowed the negative integer and asked the user to re-input a number
4	Inputting a character in the user input Question 2	How many night time minuets do you use per month? z	The program should crash as I have not declared what happens if a character is input instead of an integer	The program crashed as expected.
5	Inputting a negative number in the user input Question 3	Please enter the number of megabytes used per month: -3	The program should disallow any negative integer inputs	The program disallowed the negative integer and asked the user to re-input a number
6	Inputting a character in the user input Question 3	Please enter the number of megabytes used per month: z	The program should crash as I have not declared what happens if a character is input instead of an integer	The program crashed as expected.
7	Testing that all of the account details for all three accounts are output to the user	Once I have completed the user input it should display all 3 account details	The program should display the details of all three accounts	The program displayed the accounts as expected
8	Testing that the account information is correct when printed out to the user	Once user input is completed I shall be looking at if the information about each account is correct.	The program should display the information correctly	The program displayed the information correctly and all the information was accurate and correct.
9	Testing that the calculation to get total daytime costs is correct	Q1 input – 45 Q2 input – 23 Q3 input – 56 I shall be using the test inputs above to see if the total daytime cost is correct	The program should calculate the total correctly	The program calculated correctly and recommended me the bronze account

T.Coupe@student.liverpool.ac.uk

Thomas Coupe, 201241037, Comp 122

Pseudocode – Part 2

BronzeAccount.java

```
PUBLIC CLASS BronzeAccount EXTENDS StandardAccount
    PUBLIC BronzeAccount(INT dt, INT nt, INT mbpm)
        daytimeMinuetspermonth = dt
        nighttimeMinuetsPerMonth = nt
        megabytesPerMonth = mbpm
        packageCost = 36.00
        costOfDaytimeCalls = 0.12
        costOfEveningCalls = 0.05
        numberOfChannels = 60
        broadband = 500
        broadbandCost = 0.02
        totalCost = 0
        Spotify = false
        musiconDemand = false
        packageName = “Bronze”
```

SilverAccount.java

```
PUBLIC CLASS SilverAccount EXTENDS StandardAccount
    PUBLIC SilverAccount(INT dt, INT nt, INT mbpm)
        daytimeMinuetspermonth = dt
        nighttimeMinuetsPerMonth = nt
        megabytesPerMonth = mbpm
        packageCost = 46.00
        costOfDaytimeCalls = 0.12
        costOfEveningCalls = 0.00
        numberOfChannels = 130
        broadband = 1000
        broadbandCost = 0.01
        totalCost = 0
        Spotify = true
        musiconDemand = false
        packageName = “Silver”
```

GoldAccount.java

```
PUBLIC CLASS GoldAccount EXTENDS StandardAccount
    PUBLIC GoldAccount(INT dt, INT nt, INT mbpm)
        daytimeMinuetspermonth = dt
        nighttimeMinuetsPerMonth = nt
        megabytesPerMonth = mbpm
        packageCost = 66.00
        costOfDaytimeCalls = 0.00
        costOfEveningCalls = 0.00
        numberOfChannels = 230
        broadband = 1520
        broadbandCost = 0.01
        totalCost = 0
        Spotify = true
        musiconDemand = true
        packageName = “Gold”
```

T.Coupe@student.liverpool.ac.uk

Thomas Coupe, 201241037, Comp 122

AccountUser.java

```
PUBLIC CLASS AccountUser
    MAIN(String[] args)
                INPUT = new SCANNER(system.in)
                 INT daytimeMinuetsPerMonth;
                 INT nighttimeMinuetsPerMonth
                 INT megabytesPerMonth
                 WHILE true
                     OUTPUT (“Please enter number of daytime minuets used per month”)
                     daytimeMinuetsPerMonth = input.nextint()
                     IF daytimeMinuetsPerMonth < 0
                         OUTPUT (“You cant have a number less than zero”)
                     ELSE
                         BREAK
                 WHILE true
                     OUTPUT (“Please enter number of night time minuets used per month”)
                     nighttimeMinuetsPerMonth = input.nextint()
                     IF nighttimeMinuetsPerMonth < 0
                         OUTPUT (“You cant have a number less than zero”)
                     ELSE
                         BREAK
                 WHILE true
                     OUTPUT (“Please enter number of megabytes used per month”)
                     megabytesPerMonth = input.nextint()
                     IF megabytesPerMonth < 0
                         OUTPUT (“You cant have a number less than zero”)
                     ELSE
                         BREAK
                 BronzeAccount newBronzeAccount;
                newBronzeAccount = new BronzeAccount(DaytimeMinuetsPerMonth,
nighttimeMinuetsPerMonth, megabytesPerMonth)
                 OUTPUT ()
                 DOUBLE BronzeAccountTotal = newBronzeAccount.accountType()

                 SilverAccount newSilverAccount;
                newSilverAccount = new SilverAccount(DaytimeMinuetsPerMonth,
nighttimeMinuetsPerMonth, megabytesPerMonth)
                 OUTPUT ()
                 DOUBLE SilverAccountTotal = newSilverAccount.accountType()

                 GoldAccount newGoldAccount;
                newGoldAccount = new GoldAccount(DaytimeMinuetsPerMonth,
nighttimeMinuetsPerMonth, megabytesPerMonth)
                 OUTPUT ()
                 DOUBLE GoldAccountTotal = newGoldAccount.accountType()

        IF BronzeAccountTotal < SilverAccountTotal && BronzeAccountTotal < GoldAccountTotal
            OUTPUT (“Bronze account is the cheapest option”)

        IF SilverAccountTotal < BronzeAccountTotal && SilverAccountTotal < GoldAccountTotal
            OUTPUT (“Bronze account is the cheapest option”)

        IF GoldAccountTotal < SilverAccountTotal && GoldAccountTotal < BronzeAccountTotal
            OUTPUT (“Bronze account is the cheapest option”)
```

T.Coupe@student.liverpool.ac.uk

Thomas Coupe, 201241037, Comp 122

StandardAccount.java

```
PUBLIC ABSTACT CLASS StandardAccount
    PROTECTED DOUBLE packageCost
    PROTECTED DOUBLE costOfDaytimeCalls
    PROTECTED DOUBLE costOfEveningAndWeekendCalls
    PROTECTED INT numberOfChannels
    PROTECTED INT broadband
    PROTECTED DOUBLE broadbandCost
    PROTECTED DOUBLE totalCost
    PROTECTED BOOLEAN spotify
    PROTECTED BOOLEAN musicOnDemand
    PROTECTED STRING packageName
    PROTECTED INT daytimeMinuetsPerMonth
    PROTECTED INT nighttimeMinuetsPerMonth
    PROTECTED INT megabytesPerMonth
    PROTECTED DOUBLE overallTotalCost

PUBLIC DOUBLE AccountType()
    DOUBLE v1,v2,v3

    v1 = costOfDaytimeCalls * daytimeMinuetsPerMonth
    v2 = costOfEveningandWeekendCalls * nighttimeminuetsPerMonth
    v3 = broadbandCost *MATH.MAX(0, megabytesPerMonth-broadband

    OUTPUT ("Package name: "+packageName);
    OUTPUT ("Package cost: "+packageCost);
    OUTPUT ("Cost of daytime calls: "+costOfDaytimeCalls);
    OUTPUT ("Cost of evening and weekend calls: "+costOfEveningAndWeekendCalls+"/min");
    OUTPUT ("Number of channels: "+numberOfChannels);
    OUTPUT ("Broadband included: "+broadband+"Mb");
    OUTPUT ("Broadband cost (Above included limit): "+broadbandCost+"/Mb");
    OUTPUT ("Total daytime calls cost: "+v1);
    OUTPUT ("Total evening calls cost: "+v2);
    OUTPUT ("Total (Extra) broadband cost: "+v3);
    OUTPUT ("Total cost: "+overallTotalCost);
              RETURN overallTotalCost;
```

Additional Questions

Part 1

Question C

To calculate the average I needed to get all of the possible dice rolls from three fair dice. I used a triple for loop that used 3 integer values; var1, var2 and var3 that went from 1 to 6 and incremented by one each time it looped. As I had 3 values with 6 being the total amount of numbers on each die roll, the loop will have a total possible outcome of die rolls equal to 6 x 6 x 6 which is equal to 216. This means that there are 216 possible die outcomes with three fair dice. I then used the .getpoints() from my ThreeDiceScorer.java and assigned it to var1, var2 and var3. I then used the points variable to store this and to get the total average I divided points by 216 which gave me an average of: 24.94444444444443

T.Coupe@student.liverpool.ac.uk

Thomas Coupe, 201241037, Comp 122

Question D

In question D we were asked to consider a scenario where one of the dice were unfair and had the values:
(2,3,4,5,6,6)
From this we were asked to consider what the average would be now. Well to do this I basically reused the code that I had used in question C, however I needed to slightly change the for loop in order for it to calculate the average. I created a list with the values of the unfair die and I assigned them to a variable "badDice". I then left the first 2 for loops exactly the same as two of the other dice still had the same values. However on the 3rd for loop I needed to slightly adjust my code due to the fact that one of the dice was now unfair. As I had made a list with the unfair dice values I used an index from 0 – 5 that would be read from the list as 2,3,4,5,6,6. As I have used an index based upon my list the index values 0,1,2,3,4,5 would be read as 2,3,4,5,6,6. I then made a call to my ThreeDiceScorer again, however when I assigned the ThreeDiceScorer to var3 I needed to let the program know that var3 was the badDice so I used badDice[var3] in the call to the .getpoints() in ThreeDiceScorer. The average output as:
25.777777777778

T.Coupe@student.liverpool.ac.uk