Objektorienterad programutveckling och datastrukturer – OOAD

WS4 – Fyra I Rad

1. Är det nödvändigt att representera olika spelartyper eller hade det varit tillräckligt med en spelartyp?

Genom att dela in det i olika spelartyper kan man låta de olika spelarna ha sama namn på metoder men vad de gör är är specialiserat för spelartypen, t.ex. makeNewMove():int.

2. Vad är skillnaden mellan spelarna?

Skillnade mellan splarna är hur de gör sina drag. Exempelvis gör ComputerPlayer sitt drag genom att exekevera en algoritm vilket är väldigt annorlunda för hur HumanPlayer gör sitt drag.

Deras konstruktorer är också annorlunda då de olika spenarna behöver veta om olika saker. T.ex. så behöver inte HumanPlayer känna till något om ip-address eller port vilket är nödvändigt för RemotePlayer.

3. Vad kallas det när valet av implementation görs dynamiskt vid körning och inte vid kompilering?

Dynamiskt avsändande(dynamic dispatch).

4. Hur används detta dynamiska val i spelet och hur fungerar det?

Ovan beror makeNewMove():int på vad man har valt för olika spelartyper.

5. Vad kallas principen som används tillsammans med det dynamiska valet?

Polymorfism (de olika objekten implementerar liknande gränssnitt).

6. Användargränssnittet och modellen är separerad från varandra till olika klasser, vilka är dessa klasser?

Användargränssnittet är MainWindow.java(och relateade klasser) medan modellen beskrivs av Board.java.

7. Vad kallas det designmönster som bland annat förespråkar separering mellan UI och modell?

Model-View-Controller.

8. Varför måste ComputerPlayer ha en association till Board klassen?

När ComputerPlayer beräknar sitt drag måste den ha en uppfattning om tillståndet av Board, dvs. i vilka celler det finns lada brickor.

9. I Controller klassen skapas det en gameThread, var står denna tråd stilla och väntar på svar?

Inne i metoden playGame():void står tråden och väntar på ett giltigt drag vid currentPlayer.makeNewMove() (Sedan var detta är mer exakt beror på viken spelartyp det är som är currentPlayer).

10. Hur skulle man kunna lösa uppdelningen på olika trådar, om man inte väljer en tråd som lever under hela spelomgången?

Eftersom Swing-tråden kör under hela programmet ”skulle” man kunna lägga anropen på makeNewMove():int där.

11. Varför är det en riktigt dålig lösning att göra allt arbete på UI tråden?

När arbete görs i UI tråden kan inte uppdateringar, t.ex. knapptryckingar, hanteras vilket gör att användargränssnittet blir stillaståendes tills UI tråden har arbetar färdigt med vad den nu skulle göra.

12. Vilken information skickas över nätverket mellan två olika datorer som kör spelet?

Det skickas heltal(int) över nätvärket enligt Client.read():int samt Client.write(int):void.

Viktor Holmelin AD8746

Jelena Novacic AF6727

Jonas Eiselt AF6090