

Inlämningsuppgift 1 – Tärningsspel i java

VG del

Flera spelare

Första kravet var att man skulle kunna vara flera spelare, jag lade in en begränsning på 2-5 spelare. Detta för att jag fick problem med att arrayerna inte blev tillräckligt långa om jag inte begränsade antalet deltagare.

```
System.out.print("Hur många spelare skall det vara (2-5p)? ");
int deltagare = scan.nextInt();

if(deltagare > 5 || deltagare < 2){
    System.out.print("Ni kan bara välja mellan 2-5 spelare? ");
    deltagare = scan.nextInt();
}
```

High Score lista

```
// Bygger upp alla element till highScore samt sätter namnen till null
int forstaPlats[] = new int[2]; forstaPlats[0] = spelarId[6];
int andraPlats[] = new int[2]; andraPlats[0] = spelarId[6];
int tredjePlats[] = new int[2]; tredjePlats[0] = spelarId[6];
int fjardePlats[] = new int[2]; fjardePlats[0] = spelarId[6];
int femtePlats[] = new int[2]; femtePlats[0] = spelarId[6];

// Nedan används för att jämföra vinnarens resultat med highscore elementen
int resultatSummaSpelare = 0;
int resultatSpelarId = 0;
```

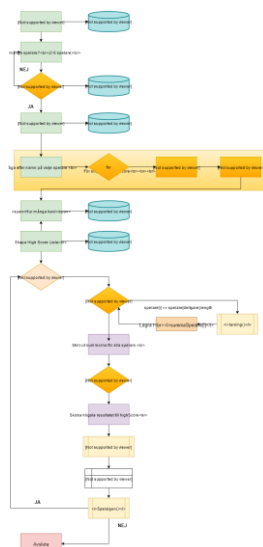
Då jag ville ha en formaterad och lite snyggare High Score stälde de krav på att kunna sortera 2 arrayer simultant. Då man inte kan blanda **integer** och **strings** samt att det tydligen inte fanns någon smidig sortering baserat på det vi lärt oss hittills så fick jag söka en annan lösning än bubblesort. Lösningen ses på bilden, jag skapade en array för varje placering med 2 element i varje array. På position [0] sparar jag id på den spelare som vunnit t.ex. Anders, detta id har på motsvarande plats i spelarNamn arrayen namnet Anders lagrat. T.ex. kanske Anders är sparad på pos 2 i SpelarNamn arrayen, på samma pos2 ispelarId finns siffran 2 lagrad.

Anders slog högsta resultatet i High Score Listan!!!

```
+-----+-----+
+      High Score      +
+-----+-----+
| Namn                | Poäng |
+-----+-----+
| Anders              | 22  |
| null                | 0   |
| null                | 0   |
| null                | 0   |
| null                | 0   |
+-----+-----+
Vill ni spela igen? (j/n):
```

Så när någon vinner en omgång så jämför jag först för att se om summan räcker för att ta sig in på listan. Så om tex spelare 2 vinner och slår bästa resultatet på highscore listan så flyttar jag först ner de andra resultaten, sedan lagrar jag spelare tvås id i pos noll i forstaPlats arrayen och sedan summan t.ex. 22 på pos ett i förstPlats arrayen.

När jag sedan skriver ut namn och summa i highscore listan gör jag det genom följande spelarNamn[forstaPlats[0]] + forstaPlats[1] osv. På pos[0] i forstaPlats finns id 2 lagrat till Anders och på pos[1] finns summan 22 lagrad. Det ser ut som på bilden till höger.



Till att börja med försökte jag lösa uppgiften med tvådimensionella arrayer men fick förkasta det efter en hel del felsökning. Skulle jag nu göra om programmet från grunden så hade nog en tabell med 2 dimensionell array varit den snyggare lösningen som gett mindre och mer lättläst kod.

Arbetet med programmet

Jag la upp arbetet genom att först göra ett flödesschema där jag försökte få med hela processen och i vilken ordning jag skulle göra de olika delarna av programmet, detta gjorde jag i draw.io. En liten överblick över huvudprogrammet syns här till höger, filerna finns bifogade i foldern "Flödesschema för programmet" på github där det blir tydligare överblick.