

Dokumentering för DiceGame™

Skriven av: Fredrik ekebro



Die.java

Denna klass representerar en tärning.

Tärningar skapas med hjälp av konstruktorn

```
Die(int sides)
```

Parametern `sides` talar om hur många sidor tärningen har.

Klassen har publika metoder:

```
int getSides()
```

Returnerar tärningens sidor.

```
int getValue()
```

Returnerar resultatet ut av tärningskastet.

```
void roll()
```

Kastar tärningen mha `java.util.Random` som sätter värdet på tärningskastet mellan 1 och antalet sidor.

Player.java

Denna klass representerar en spelare i tärningsspelet. Spelaren har ett namn och ett antal tärningar som spelaren kan kasta. Efter ett tärningskast kan spelet öka spelarens poäng.

Spelare skapas med konstruktorn

```
Player(String name)
```

Parametern `name` ger spelaren ett namn.

Klassen har publika metoder:

```
String getName()
```

Returnerar spelarens namn.

```
void rollDice()
```

Rullar tärningen eller tärningarna.

```
int getDieValue()
```

Tar tärningens nummer.

```
int getScore()
```

Returnerar spelarens totala poäng.

```
void increaseScore()
```

Ökar spelarens poäng med 1.
`void increaseScore(int score)`
Ökar spelarens poäng med "score" variabeln.
`void addDie()`
Lägger till en ny tärning som har deklarerade sidor.

EasyDiceRoller.java

Detta är ett program för tärningsspel där spelaren ska gissa vad tärningens nummer kommer bli, om spelaren lyckas gissa rätt så ökar deras poäng med ett. Innan spelet börjar bestämmer spelaren hur många sidor(`sides`) tärningen har och hur många omgångar(`rounds`) som ska spelas. Programmet använder `Player` klassen som använder sig av `Die` objekt för att modellera tärning/tärningarna.

Programmet kommer ställa frågor som användaren svarar på mha `java.util.Scanner`, frågorna ställs är i ordern:

1. `How many rounds?`
2. `How many sides?`
3. `What is your name?`

Efter det så körs den faktiska koden inom en `for` loop.

```
for (int i = 0; i < rounds; i++)
```

Loopar hur många rundor som var matat in av användaren. För varje runda så kastas tärningen och innan numret visas så frågas spelaren:
`What number will be displayed on roll (#)?`.
Användaren gissar och gissar dem rätt så ökas `Score` med ett.
Om man gissar fel så händer inget och det fortsätter till nästa runda utan att få någon poäng.

Den loopar frågan tills gränsen av rundor har nått och sen matar ut hur många rätt svar spelaren fick och programmet stänger av sig själv efteråt.

EasyDiceRoller_Multiplayer.java

Detta är en annan version av tärningsspel där fler än en spelare körs. Detta ingår att bestämma hur många spelare ska kasta tärning/tärningar. Deras tärningskast läggs till sammanlagt varsitt totalt och den med den högsta totalen är deklarerat vinnaren för spelet.

Programmet kommer ställa frågor som användaren svarar på mha `java.util.Scanner`, frågorna ställs i ordern:

1. `How many players?`
2. `How many dice?`
3. `How many sides on the die?`
4. `How many rounds?`

Skulle någon av dessa frågor få ett värde mindre än '1' så skriver programmet "Insufficient Input! DICE GAME™ shutting down...", sen stänger den av sig själv.

Efter allt det så körs programmet och med att ställa frågan vad spelarens namn är för varje spelare. Efter att alla spelare har skrivit in sina namn så kastar spelarna sina tärningar per runda och adderar till deras total. Som ett exempel med två spelare för två rundor:

On Round 1:

```
{Player1Example}, You Scored: 12  
{Player2Example}, You Scored: 14
```

On Round 2:

```
{Player1Example}, You Scored: 12  
{Player2Example}, You Scored: 17
```

| FINAL SCORE |

```
{Player1Example}, Your Total Score: 24  
{Player2Example}, Your Total Score: 31
```

```
{Player2Example} You win!
```

Spelet därefter stänger av sig själv.

MasterDiceRoller

Detta är ett bonus tillägg som jag gjorde på Jswing som är inte klar ännu men dess grundläggande funktioner fungerar.