

Dice game

Generellt

Spelet är byggt i Java och innehåller tre olika klasser, **Die** dvs tärningen i spelet, **player** som är spelare och **DiceGame** som är spelet.

Die

Variabler

Tre variabler har använts, `private int value`, `private int numberOfSides` och `private Random` slumpgenerator

- `private int value` - för att lagra värdet på tärningen
- `private int numberOfSides` - för antalet sidor på tärningen
- `private Random` slumpgenerator - för att få fram slumpmässiga tal

Metoder

- `"public void roll()"` - har använts för att "kasta" tärningen och sedan ge den ett slumpmässigt tal.
- `"public Die(int numberOfSides)"` - är konstruktorn, hämtar i detta fall antal sidor på tärningen.
- `"public int getValue"` - här har en getter-metod använts för att kunna hämta värdet på tärningen efter varje omgång av kast.

Player

Player-klassen representerar spelaren och består av tre variabler; namn, poängen och tärningen.

Variabler

- `private String name` - för att lagra namnet på den som spelar
- `private int score` - för att lagra spelarens poäng
- `public Die die` - är spelarens tärning.

Konstruktorn `"public Player(String name, int numberOfSides)"` hämtar namnet och antal sidor på tärningen som spelaren har skrivit in och startar spelet med 0 poäng.

Metoder

- `public void rollDice()` – har använts för att ”rulla” tärningen
- `public int getDieValue()` – för att ge tärningen ett tal (värde)
- `public void increaseScore()` – för att öka spelarens poäng med 1 i de fallen spelaren gissar rätt.
- `public void setScore` – används för att sätta spelarens korrekta poäng.
- `public void addDie(int numberOfSides)` – för att skapa och lägga till en ny tärning till spelaren.

DiceGame

DiceGame klassen innehåller ”main-metoden” vilket gör att det är härifrån själva spelet spelas.

Variabler

- `Scanner scanner = new Scanner(System.in);` - har använts för att kunna ta emot input från spelaren via terminalen.
- `Player = new player(name, numberOfSides);` - variabeln `name` lagrar namnet spelaren matar in, `numberOfSides` lagrar värdet för antal sidor som angetts av spelaren.
- `Int rounds = scanner.nextInt();` - för att kunna ange hur många omgångar spelaren vill spela spelet.

Metoder

- `Public static void main(String[] args)` – main metoden
- `System.out.println` – för att skriva ut händelser
- For loop :
 - `For(int round = 1; round <= rounds; round++)` – för att kunna skriva ut hur många omgångar som ska spelas
 - `If (guess == dieValue)` = om värdet man har gissat på är det samma som tärningsvärdet får man 1 poäng.
 - `else` = annars, (om man gissar fel) får man inget poäng.

Motiveringar:

Variabler och metoder som har använts har varit nödvändiga för att skapa ett fungerande tärningsspel. Variablerna och metoderna har gjort det möjligt för spelet att lagra spelarens namn, antal omgångar som ska spelas, antal sidor tärningen ska ha samt spelarens poäng under spelets gång men även spelarens slutpoäng.

Hur man spelar DiceGame

- Spelaren börjar med att fylla i sitt namn, antal omgångar som ska spelas och hur många sidor man vill ha på tärningen.
- Programmet kommer med hjälp av informationen att skapa tärningen och spelaren.
- Spelaren gissar på tärningsvärdet, i det fall spelaren gissar rätt får spelaren ett poäng.
- I det fall spelaren har valt att spela mer än en omgång ber spelet spelaren att gissa igen.
- När spelaren har spelat klart sina omgångar informeras spelaren om att spelet är slut och får även sin slutpoäng.