

Uppgift 2 – Nm

Författare:
Alexander Lindgren
alli1660

Inlämningsuppgift nr.2
Datum: 22/02-2024
JDK 11

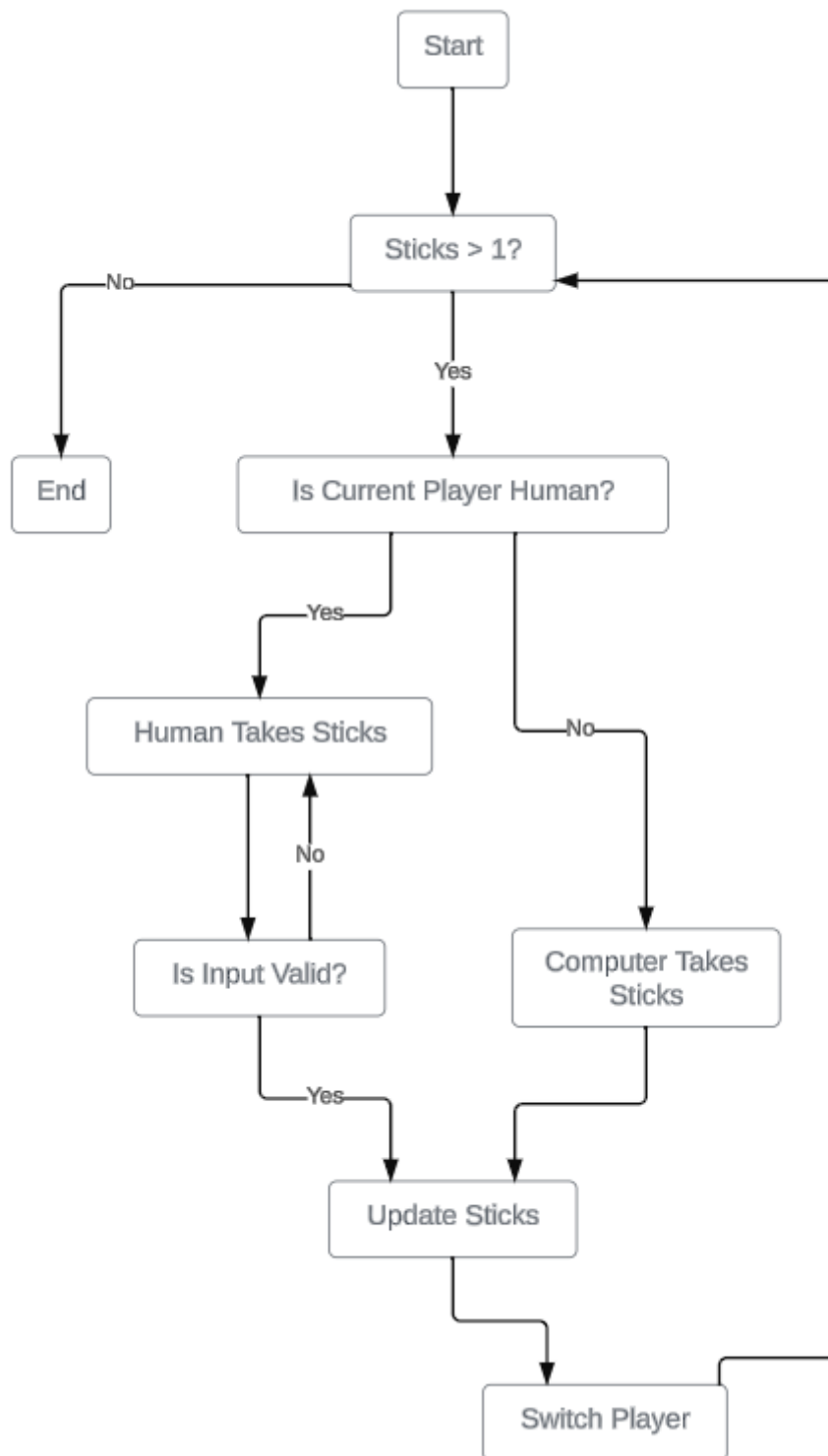
Regler

Spelet spelas med minst 10 tändstickor. Vid spelets start väljer spelaren antalet tändstickor som ska användas. Det finns två spelare i spelet: en mänsklig spelare och en datorspelare. Spelet börjar med den mänskliga spelaren. I varje runda får spelaren ta ett antal tändstickor från en gemensam hög. Antalet tändstickor som kan tas under varje drag är mellan 1 och hälften av det totala antalet tändstickor som finns kvar i högen. Spelet fortsätter tills det endast finns en tändsticka kvar i högen. Den spelare som tvingar motståndaren att ta den sista tändstickan förlorar.

Strukturer och metoder

Spelet startar i Main-metoden (Main-klassen): Användaren ombeds att mata in antalet tändstickor som ska finnas med i spelet. Inmatningen valideras för att se till att det angivna antalet är minst 10. Jag har valt att sätta en gräns på minst 10 stickor för att säkerställa att inte negativa tal eller väldigt låga siffror som exempelvis 1 ska kunna spelas.

Ett objekt av typen HumanPlayer och ComputerPlayer skapas. Ett objekt av typen NmGame skapas med de skapade spelarna och antalet tändstickor. Spellogik (NmGame-klassen): När spelet startar, börjar spelet med antalet tändstickor och de två spelarna. Spelet körs med play-metoden. Under spelets gång alternerar spelarna och gör sina drag. Antalet tändstickor minskar efter varje drag. Spelet fortsätter tills endast en tändsticka återstår, detta görs med hjälp av en while-sats.



Datorns beteende: När det är datorns tur i spelet, beräknar den antalet pinnar den ska ta. Detta görs genom att dividera det aktuella antalet pinnar kvar på bordet med 2 och avrunda till närmaste heltal. Detta ger datorn en strategi att alltid ta så många pinnar som möjligt för att minimera antalet pinnar kvar för motståndaren.

Människans beteende: När det är människans tur blir den ombad att mata in hur många stickor som ska tas ur högen. Sedan görs det en kontroll för att de om inmatningen är korrekt för programmet att fortsätta. Mer exakt om hur kontrollen går till finns längre ner vid felhanterings avsnittet.

Standardbibliotek

Några Standardbibliotek som används i programmet, `java.util.Scanner`: Används för att läsa inmatning från användaren i `Main`-klassens `main`-metod och `NmGame`-klassens `play`-metod.

`java.lang.System`: Används för att skriva ut meddelanden till konsolen i olika delar av koden, såsom när spelaren ombeds att välja antal pinnar och när datorn informerar om sitt drag.

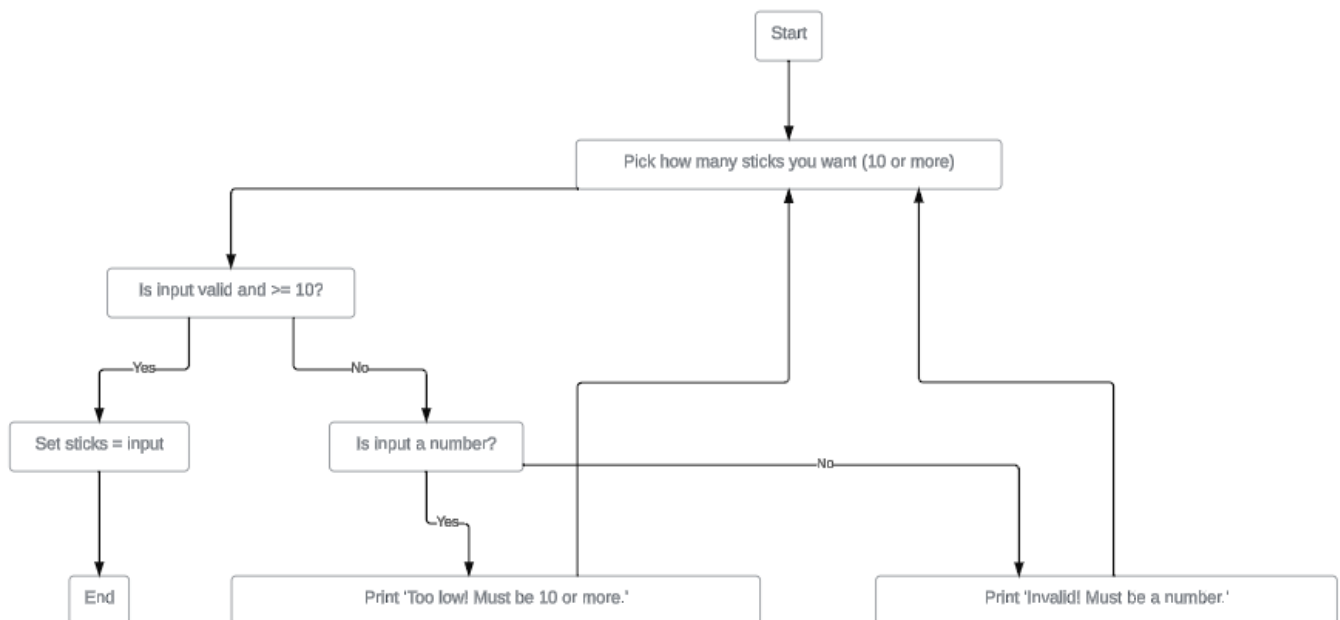
`java.lang.String`: Används för att utföra operationer på inmatade strängar, såsom att trimma whitespace och matcha mönster i `Main`-klassens `main`-metod.

`java.lang.Integer`: Används för att konvertera strängrepresentationer av heltal till faktiska heltal, och för att jämföra heltal i `Main`-klassens `main`-metod.

`java.lang.NumberFormatException`: Används för att hantera undantag som kan uppstå när du försöker konvertera en sträng till ett heltal som inte är ett giltigt heltal i `Main`-klassens `main`-metod.

Felhantering

Inmatningen av heltal från en användare kan skapa många möjligheter för programmet att krascha, för att förhindra detta måste vi göra kontroller som ser till att rätt värden mattas in. Första inmatningen sker i början av spelet, där det frågas om hur många stickor det ska spelas om, här används en `while`-loop som kommer att fråga användaren om ett giltigt värde så länge det krävs, se flödesdiagrammet nedan.



Problemen som kom upp vid inmatning var att programmet krascha vid exempelvis inmatningar av bokstäver och symboler.

För att lösa detta gör jag att inputsen till en sträng(`String input = scanner.nextLine().trim();`) där `nextLine()` och `trim()` läser första raden och trimmar bort mellanslag, detta för jag vill att det inte ska spela någon roll om användaren råkar använda ett mellanslag vid inmatningen.

Användarinputen kontrolleras för att säkerställa att den är ett giltigt heltal och att det valda antalet pinnar är minst 10. Detta görs genom användning av en reguljär uttryckssträng (`matches("\\d+")`) för att kontrollera att inmatningen endast innehåller siffror. Sedan görs strängen om till En integer med hjälp av metoden `Integer.parseInt()` som gemför heltalsvärdet med det värde som behövs, 1 mellan antal stickor / 2 eller vid första inmatningen av spelets stickor som måste var 10 eller över.