

Obligatorisk uppgift 2 - 1DL028

Alhassan Jawad

February 16, 2024

1 Klasser och Metoder

1.1 NmGame Klass

`NmGame`-klassen är ansvarig för att köra spelet. De viktigaste metoderna inkluderar:

- `main`: Startar spelet genom att initialisera antalet pinnar och anropa `startGame`.
- `startGame`: Hanterar spelloopen och alternerar mellan mänskliga och datorspelare.
- `makeMove`: Anropar nuvarande spelares `makeMove`-metod.
- `endGame`: Avgör vinnare baserat på antal stickar kvar.

1.2 Player Klass

`Player`-klassen är en abstrakt klass med metoderna:

- `makeMove`: En abstrakt metod som ska implementeras av subclasser.
- `getName`: Returnerar namnet på spelaren.

1.3 ComputerPlayer Klass

`ComputerPlayer`-klassen representerar datorspelaren och implementerar `makeMove`-metoden för att göra ett slumpmässigt drag.

1.4 HumanPlayer Klass

`HumanPlayer`-klassen representerar den mänskliga spelaren och implementerar `makeMove`-metoden för att interaktivt ta bort pinnar.

2 Designbeslut

2.1 Utvecklingsprocess

Först började jag med att programmera spelet utifrån sticks21 spelet där man har en konstant initial hög av 21 pinnar och får endast ta 1-3 pinnar. Dessutom så började jag programmera spelet utan att använda mig av polymorfism, dvs jag programmerade allt i en och samma klass. Efter att ha programmerat klart allting och spelet funkade så skrev jag om koden så att den använder sig av polymorfism, låter användaren bestämma hur många pinnar det ska finnas från början och låta spelarna ta $1 - \frac{sticks}{2}$ pinnar per tur.

2.2 Användning av Standardbibliotek

Den enda standardbibliotek jag använde mig av var util modulen som innehåller klassen `Scanner`. Den importerade jag för att hantera användarinmatning, vilket gav mig ett robust och standardiserat sätt att interagera med konsolen.

2.3 Gränssnitt för Klasser

Gränssnitten för de olika klasserna är noggrant utformade för att säkerställa en tydlig och konsekvent interaktion. Det sträcker sig bortom att bara definiera metoder; det inkluderar även specifikationer om indataförhållanden och antaganden om metदानrop.

2.3.1 Player-klassen

Gränssnittet för `Player`-klassen specificerar:

- `makeMove`: Den definierar protokollet för att göra ett drag.
- `getName`: En metod för att hämta spelarens namn. Så `getName` returnerar "Human" för den mänskliga spelaren och "Computer" för datorspelaren.

2.3.2 ComputerPlayer-klassen

Den implementerar metoden `makeMove` för att göra ett slumpmässigt drag.

2.3.3 HumanPlayer-klassen

Den implementerar metoden `makeMove` för att tillåta interaktiv borttagning av pinnar och säkerställa en giltig inmatning inom specificerade villkor.

2.3.4 NmGame-klassen

Det är i denna klass som spelet startas med `startGame` och avslutas med `endGame`.

3 Implementering av Spelregler

Implementeringen av spelregler, som att kontrollera om ett drag är giltigt, hanteras främst i `HumanPlayer`-klassen. Användaren uppmanas att ange antalet pinnar att ta bort, och programmet ser till att det är ett giltigt drag. Om ett ogiltigt drag görs kommer felmeddelanden att skrivas ut innan programmet avslutas.

Den enda antagandet som jag använde mig av är att man får inte börja spela `NmGame` om man har inte minst 2 pinnar i högen (men detta är inte riktigt ett antagande då spelreglerna säger så).