

Střední průmyslová škola elektrotechnická

Informační technologie

Praha 2, Ječná 30

Card game

Turcsányiová Veronika

Informační a komunikační technologie

2025

Obsah

1	Cíl práce.....	3
2	Popis hry.....	3
2.1	Algoritmus.....	3
2.2	Postavy.....	3
2.3	Mechanika.....	3
3	Systém requirements.....	3
4	Základní struktura.....	3-4
5	Testovací data.....	4
6	Uživatelská příručka.....	4
7	Závěr.....	4
8	Zdroje.....	4

1 Cíl práce

Cílem hry je vytvořit jednoduchou konělovou hru Prší pro jednoho hráče proti počítači. Hráč zadává příkazy do konzole, program vypíše a splní daný příkaz. Hra by neměla být složitá a měla by mít všechny funkce jako hra Prší.

2.1 Algoritmus

Program umožňuje hráči hrát a vstoupit kartu, uložit a znovu načíst hru, zobrazit si karty a manuál a ukončit hru. Jako první se hráči a počítači rozdají 4 startovní karty, pak se nastaví karta na vrchu balíčku a aktuální typ a barva karty. Pokaždé co se hráč pokusí zahrát kartu, tak se kontroluje, jestli danou kartu má a jestli se dá položit. Následně počítač reaguje buďto zahráním další karty, nebo tažením karty z původního balíčku. Pokud hráč zahrane Jacka (J), dostane na výběr, na jakou barvu chce změnit. Ve stejném případě si počítač spočítá, jaké barvy má nejvíce a na tu aktuální barvu změní. Pokud je zahrána Sedm (SEVEN), další na řadě dostane 2 karty navíc. Pokud je zahráno Eso (A), další na řadě toto kolo stojí. Když hráč zahrane Pikového Krále (DIAMONDS K), dostane možnost vstoupit si 1 kartu navíc. Počítač má v tomto případě 50% šanci že si kartu navíc vezme. Hra se ukončí, když počítač nebo hráč nemají žádné karty, když hráč uloží hru nebo zadá příkaz Exit.

2.2 Postavy

Jediné 2 postavy v této hře jsou Player, což je hráč samotný, a Computer, což je jednoduchý algoritmus rozhodování.

2.3 Mechanika

PlayACard – Příkaz na zahrání karty.

DrawACard – Příkaz na tažení karty z původního balíčku.

SaveGame – Příkaz pro uložení aktuálních hodnot hry do souboru. Tento příkaz navíc hru ukončí.

LoadGame – Příkaz pro načtení hodnot z uloženého souboru.

SeeManual – Příkaz pro vypsání příručky.

GetPlayerPack – Příkaz pro vypsání karet, které hráč má.

Exit – Příkaz pro ukončení hry.

3 Systém requirements

Program byl vyvíjen v jazyku Java, IntelliJ IDEA verze 2024.2.1. Není potřeba stahovat žádné externí knihovny. Program je možné spustit v prostředí podporujícím Javu.

4 Základní struktura

Program je objektový a je rozdělen do několika tříd a podtříd. Mezi nejdůležitější třídy patří:

Player – Obsahuje metody a logiku pro hráče, kontroluje, jestli může hráč zahrát kartu. Rozděluje karty do playerPack a vrací do cardPack

Computer – Obsahuje metody a jednoduchou rozhodovací logiku pro počítač. Pokud může, hraje kartu, která se následně vrátí do cardPack.

Pack – Třída pro hlavní balíček karet, obsahuje metody pro rozdání startovních karet, inicializování vrchní karty balíčku, metodu pro Sedm (SEVEN) a Eso (A)

Cards1 – Třída přečte karty z textového souboru, vloží je do listu a zamíchá je. Metoda se používá v Pack, kde se karty přidají do cardPack.

Console – Třída přijímá příkazy, rozděljuje a zpracovává je. Kontroluje, jestli není gamelsOver.

Game – Třída inicializuje všechny objekty, nastaví aktuální barvu a typ karty u Computer, Player a Pack. Metoda Play() zajišťuje běh hry a opětovné nastavování barvy a typu. Kontroluje, jestli někdo nevyhrál.

Main – Spouští hru.

5 Testovací data

Doporučuji program otestovat vykoušením všech příkazů a následně zkoušet dát pár špatných vstupů. Tak můžete zjistit, jestli program udělá to, co má a nespadne. Já testovala špatné zadávání, příkazy, speciální karty a reakce Computer. Dobré by bylo otestovat, když je vrchní karta cardPack sedm (SEVEN) nebo Eso (A) jestli se metoda opravdu spustí pouze jednou a nebude se opakovat, dokud se nemění karta.

6 Uživatelská příručka

Program se ovládá zadáváním příkazů do konzole. Střídavě s počítačem hráč hraje nebo táhne kartu. Hra pokračuje, dokud někdo nebude mít žádné karty, nebo dokud hráč hru neukončí. V programu je i možnost vypsát Player's manual, ve kterém jsou popsány příkazy a karty.

7 Závěr

Jeden problém bylo špatné nastavování aktuální barvy a typu karty, kdy se sice nastavili v Pack ale ne v Player. Ten potom nemohl pokládat žádné karty. Ale úplně největší problém nastal při měnění barvy. Tehdy když Player nebo Computer změnili barvu, sice se změnila u nich, ale Pack ji zase vrátil na původní. Oba problémy jsem vyřešila tak, že jsem přerušila dědění z Pack do Player a Computer. Celkově mě projekt docela bavil.

8 Zdroje

[1] Ve třídě Player, v metodě playerPlayCard() mi část kódu vygeneroval Chat gpt, konkrétně `Optional<Card> optionalCard = playerPack.stream().filter() a .findFirst()`. Řádky 44-47.

[2] Ve třídě Player, v metodě playCard() je od Chat gpt `Iterator<Card>, iterator.hasNext() a iterator.next()`. Řádky 37-38.

[3] Ve třídě Cards1, v metodě readCardsFromFile() je od Chat gpt `Collections.shuffle`. Řádek 40.

[4] Ve třídě Cards1Test, v metodě testCardsFromFile(), `PrintWriter writer = new PrintWriter(file)` - JENKOV, Jakob. *Java IO: PrintWriter*. Jenkov.com [online]. 10. září 2015 [cit. 2025-05-30]. Dostupné z <https://jenkov.com/tutorials/java-io/printwriter.html>. Řádky 28-32.

`S cards.stream().anyMatch(c -> c.getType())` mi pomohl Claude ai. Řádky 38-41.