

Zápočtový program zimní semestr 2020/2021

Hra Logik – Zuzana Bohatová

Program řeší hru Logik z hlediska hráče, který hádá. Pravidla hry: Hráč A si vybere nějakou 5místnou variaci ze zadaného počtu čísel (ve variaci se mohou opakovat), dále nazývanou jen jako kód. Hráč B se snaží kód uhodnout. Hráč A každý pokus ohodnotí kombinací černých a bílých bodů, kde černý znamená, že správná barva leží na správném místě, a bílý znamená, že číslo v kódu je, ale není umístěné na správném místě.

Na začátku programu si uživatel vybere, která čísla může kód obsahovat od 0 do 9, např. pokud zadá 5, bude kód obsahovat čísla od 0 do 5, pokud 0, bude jen jedna variace a to 00000, pokud zadá 9, bude kód obsahovat čísla od 0 do 9.

Poté si zvolí, který hráč chce být. Vybere si jeden ze dvou módů:

1. MOD1 = program vybere kód a uživatel ho hádá
2. MOD2 = uživatel si vybere kód a program ho uhádne

Uživatelská část dokumentace

Uživatel potřebuje mít ke spuštění programu nainstalovanou platformu .NET Core 3.1.

Manuál pro uživatele

1. program můžete spustit třemi způsoby :
 - a. bez argumentů
 - i. spusťte program
 - ii. přečtěte si vypsaná pravidla hry
 - iii. vyberte si jaká čísla bude moct obsahovat váš kód, pokud zadá 5, bude kód obsahovat čísla od 0 do 5, pokud 0, bude jen jedna variace a to 00000, pokud zadá 9, bude kód obsahovat čísla od 0 do 9
 - iv. vyberte mód hry – zadáním čísla 1 (hádáte kód) nebo 2 (vybíráte kód)
 - b. s jedním argumentem
 - i. jako argument zadáte číslo od 0 do 9 - vyberte si jaká čísla bude moct obsahovat váš kód, pokud zadá 5, bude kód obsahovat čísla od 0 do 5, pokud 0, bude jen jedna variace a to 00000, pokud zadá 9, bude kód obsahovat čísla od 0 do 9
 - ii. pokud bude argument zadán chybně, program vypíše chybovou hlášku a skončí
 - iii. vyberte mód hry – zadáním čísla 1 (hádáte kód) nebo 2 (vybíráte kód)
 - c. s dvěma argumenty
 - i. jako argument zadáte číslo od 0 do 9 - vyberte si jaká čísla bude moct obsahovat váš kód, pokud zadá 5, bude kód obsahovat čísla od 0 do 5, pokud 0, bude jen jedna variace a to 00000, pokud zadá 9, bude kód obsahovat čísla od 0 do 9
 - ii. jako druhý argument vyberte mód hry – zadáním čísla 1 (hádáte kód) nebo 2 (vybíráte kód)
 - iii. pokud bude nějaký argument zadán chybně, program vypíše chybovou hlášku a skončí

2. pokud hádáte – stiskli jste 1:
 - a. při hádání zadáváte pěticiferná čísla obsahující čísla 0 až zadané číslo
 - b. stisknutím klávesy 1 můžete požádat o nápovědu, která se vám vypíše ve formátu např. 32---
 - c. po uhodnutí kódu nebo vyčerpání nápovědy, vám program nabídne novou hru, kterou spustíte klávesou 0
 - d. pokud zmáčknete jinou klávesu program skončí
3. pokud vybíráte kód – stiskli jste 2:
 - a. vyberte si kód
 - b. ohodnocujte odhady programu podle pravidla: černý bod = správné číslo je na správném místě a bílý bod = správné číslo, ale na špatném místě
 - c. součet vašich odhadů nemůže být záporné číslo nebo číslo větší než 5, jelikož kód má pět míst
 - d. každý jednotlivý odhad nemůže být záporný nebo větší než 5
 - e. odhad 4 černé body a 1 bílý není možný, jelikož nemůžete mít 4 čísla na správném místě a jedno správné číslo na špatném místě
 - f. až program hru uhodne, nebo pokud nezbudou žádné další možnosti, tak můžete hru spustit znovu pomocí klávesy 0, pokud hrát dál nechcete, stiskněte jakoukoliv jinou klávesu

Programátorská část dokumentace

Výběr algoritmu:

Mód 1: program vybere náhodný kód a uživatel ho hádá, program ohodnocuje jednotlivé odhady, kombinací černých a bílých bodů, uživatel má také možnost nápovědy – ukáže se mu jedno číslo na správném místě.

Mód 2: uživatel si vybere svůj kód a program ho hádá, uživatel jednotlivé odhady hodnotí, program vybírá odhady podle jednoduchého algoritmu: máme všechny existující varianty a ty postupně procházíme a srovnáváme, jestli by ohodnocení vůči nim bylo stejné jako ohodnocení uživatele, pokud ano, daný odhad se posune dopředu v možnostech, pokud ne, odhad ignorujeme a snížíme počet možností, poté vybereme náhodně odhad z možností, co nám zbyly

Program

1. Class Program – obsahuje pouze metodu main, na konzoli se vypíší pravidla hry a vytvoří se nový objekt Logic a proběhne inicializace hráčů, pokud proběhla dobře, spustí se hra, je zde zajištěn také možný restart hry
2. Class Logic – hlavní třída programu
 - a. internal bool Inicializace() – přijímá jako parametr pole argumentů, pokud pole žádné argumenty neobsahuje, požádáme uživatele o zadání nejvyššího čísla a výběru módu hry. Pokud pole obsahuje 1 argument – otestujeme, zda je ve správném formátu a pokud ano, požádáme uživatele o zadání módu hry. Pokud pole obsahuje 2 argumenty, otestujeme oba jestli jsou správně zadané. Pokud jakýkoliv z argumentů byl chybně zadán, program vypíše chybovou hlášku a vrátí false – následně program skončí. Pokud vše bylo v pořádku, vrátí true a spustí se hra.

- b. `private bool NastavNejvyssiCislo()` – uživatel zvolí, jaké nejvyšší číslo může kód obsahovat zapsáním čísla od 0 do 9, např. napíše 6 – kód bude obsahovat čísla 0 až 6, nebo se vezme zadaný mód ze vstupních argumentů programu
 - c. `private void InicializaceHracu()` – vytvoří hráče podle toho jaký byl zadán mód
 - d. `internal void ZacniHru()` – řídí celý běh hry
 - i. pracujeme s proměnnými `posledniOdhad`, `posledniReakce`
 - ii. dokud `posledniReakce` neobsahuje pět černých bodů, tak hádáme dál – spouštíme metodu `hadajiciHrac.SehrajTah()`, pokud nám ale tato metoda přepíše proměnnou `konecHry` na `true`, znamená to, že došli možnosti pro hádání (v nějakém ohodnocení byla chyba), tak hra také končí, pokud proměnná `konecHry` zůstává `false`, tak se spouští metoda `hodnoticiHrac.Vyhodnot()`, jejíž návratová hodnota se uloží do proměnné `posledniReakce`
 - iii. a cyklus začíná od začátku
 - iv. pokud cyklus skončí, tak se zkontroluje, zda vyhrál počítač nebo uživatel a vypíše vítěznou hlášku, kolik kol a kolik nápovědy potřeboval na výhru
3. `class HraciPlan`
- a. dvě `internal` proměnné: `moznosti` – pole všech možných variací a `zbyvaMoznosti` – kolik z variací může být hledaným kódem podle dosavadního hodnocení
 - b. `internal HraciPlan(nejvyssiCislo)` – konstruktor hracího plánu – nastaví počet možností (`zbyvaMoznosti`) – podle zadaného nejvyššího čísla a sestaví pole všech možností metodou `NastavMoznosti()`
 - c. `private void NastavMoznosti(nejvyssiCislo)` – nastaví pole všech možností
4. `class Hrac` – rodičovský objekt pro `HadajicihoHrace` a `HodnoticihoHrace`
- a. `protected bool automat` – zda je hráč počítač nebo uživatel
 - b. `internal Hrac(automat)` – konstruktor hrace
 - c. `internal bool Automat` – vlastnost `Automat`
 - d. `internal int[] VyberKod(nejvyssiCislo)` – vygeneruje náhodný kód
 - e. `protected int[] VyhodnotPocitace(odhad, kod)` – vyhodnotí a vrátí reakci na daný odhad vůči zadanému kódu
5. `class HadajiciHrac : Hrac`
- a. `private int[] odhad` – ukládá se sem hráčův odhad
 - b. `private bool konecHry` – jestli už hra skončila (jestli jsou ještě nějaké možnosti)
 - c. `private List napoveda` – ukládáme sem, která políčka kódu již známe z nápovědy
 - d. `private int pocetKol` – kolik kol už je odehraných
 - e. `internal bool KonecHry` – vlastnost `KonecHry`
 - f. `internal int PocetKol` – vlastnost `PocetKol`
 - g. `internal HadajiciHrac(automat)` – konstruktor `HadajicihoHrac`
 - h. `internal int GetDelkuNapovedy()` – vrací délku nápovědy
 - i. `internal int[] SehrajTah()` – vybere jeden možný odhad podle toho, jestli je hádající hráč uživatel (`ZadejSvujOdhad()`) nebo počítač (`JsouJesteNejakeMoznosti()`) a vrátí daný odhad
 - j. `private bool JsouJesteNejakeMoznosti()` – metoda pracuje s polem možností a proměnnou `zbyvaMoznosti` z objektu `hraciPlan` (odkazujeme na ně referencí) a porovnává všechny možnosti s reakcí, kterou máme a pokud by reakce neseděla tak snižuje proměnnou `zbyvaMoznosti`, ve chvíli, kdy dojdou možnosti vrací `true`

- (je konec hry), pokud nedojdou zavolá metodu VygenerujDalsiOdhad(), která náhodně vybere jeden z ještě možných odhadů
- k. private void VygenerujDalsiOdhad() – náhodně vybere jeden z možných odhadů a uloží ho do public proměnné odhad
 - l. private int[] ZadejSvujOdhad() – žádá po uživateli, aby zadal svůj odhad a kontroluje jeho syntaktickou správnost (délku, zda obsahuje správná čísla, ...), pokud uživatel místo odhadu napíše číslo 1, spustí metodu nápovědy
 - m. private void Napoveda() – postupně uživateli odkrývá další čísla kódu
6. class HodnoticiHrac : Hrac
- a. internal HodnoticiHrac(automat) – konstruktor HodnoticihoHrace
 - b. private int[] hadanyKod – uloží se sem hádaný kód
 - c. internal int[] HadanyKod() – vlastnost HadanyKod
 - d. internal int[] Vyhodnot() – vyhodnotí daný odhad vůči kódu podle toho, zda je hodnotící hráč uživatel nebo počítač, a vrátí dané vyhodnocení
 - e. private int[] VyhodnotUzivitelem() – žádá po uživateli, aby ohodnotil předchozí odhad a pak kontroluje syntaktickou správnost ohodnocení a vrátí reakci uživatele

Reprezentace vstupních dat a jejich příprava

Vstupní data jsou pětice čísel skládající se z číslic, která si uživatel na začátku vybere, když zvolí 6, kód bude obsahovat čísla 0, 1, 2, 3, 4, 5, 6. V případě, že uživatel zadá více nebo méně čísel nebo mezi čísly bude 7, 8 nebo 9, program napíše, kde se stala chyba a požádá o nová vstupní data. V případě, že vstup není číslo, program vypíše, že jste nezadali číslo a opět požádá o nový vstup. Obdobně se program chová, když uživatel ohodnocuje odhad – kontroluje, zda se zadala čísla a také jestli se zadala správná čísla od 0 do 5, jestli dávají smysl, pokud něco nesedí vypíše, kde se stala chyba a požádá o její napravení. Pokud uživatel místo svého odhadu zadá číslo 1, spustí se krok nápovědy. Pokud na konci hry zadá číslo 0, hra se spustí znovu. Vše je doprovázeno vysvětlujícími hláškami a návody.

Reprezentace výstupních dat a jejich interpretace

Na začátku hry program vypíše pravidla hry a požádá uživatele o výběr, které největší číslo může kód obsahovat. dále ho požádá o výběr módu. následně vypíše, které varianty si uživatel vybral a vybídne k začátku hry.

Program vypisuje postupně každý odhad uvxyz a jeho vyhodnocení, ve formátu:
počet černých bodů: x
počet bílých bodů: y

Pokud je některá z informací zadána chybně požádá uživatele o nová vstupní data a snaží se poradit, kde se stala chyba. Ve chvíli, kdy uhádnete kód, tak program vypíše hlášku o vítězství a na kolik pokusů hráč vyhrál a kolikrát potřeboval nápovědu. A nabídne restart hry.

Sada testovacích dat

Jednotkové testy

1. LogikTest – InicializaceTest() – Test metody inicializace, testujeme jednotlivé možnosti zadání argumentů: 2x test správně zadaných argumentů, argument 1 není číslo,

argument 2 není číslo, argument 1 je číslo větší než povolené číslo, argument 2 je větší číslo než povolené

2. HracTest – VygenerujKodTest() – test metody vygeneruj kód, testujeme, zda se ve vygenerovaném kódu nenachází číslo větší než povolené
3. HodnoticiHracTest – VyhodnotTest() – test metody vyhodnoť, testujeme, zda se reakce, kterou spočítá program rovná naší očekávané reakci.
4. HadajiciHracTest – SehrajTahTest() – test metody sehraj tah, kontroluje, zda odhad programu obsahuje jen povolená čísla

Testovací data

Špatná vybraní počtu čísel – zadání písmena nebo dvouciferného čísla

Špatné vybraní módu – zadání jiného čísla než 1 a 2 nebo zadání písmena či řetězce znaků

MOD1 – zadání kódu špatné délky, zadání písmene, 1 (náповěda), možné problematické kódy: 99999, 11111, 23333, 11311, 33336, 11124, 11135, 11125, 77734, 66667, 22221, 02333, 21733, 02282, 11139, 11333, 67333, 33366, 66888, 11331 a několik dalších zkušebních kódů: 01234, 43210, 81610, 52710, 61443, 44215, 14624, 51510, 17350.

MOD2 – Zadání písmena či špatné ohodnocení, ohodnocení, které nedává smysl (např. 4 černé body a 1 bílý), uživatel někde v ohodnocení udělá chybu – poté by program měl vypsat, že nezbyly žádné další možnosti

Po ukončení hry – 0 – test restartu hry