Domácí úkol č. 2

Zadání

Vaším úkolem je rozšířit hru Hádání čísel, kterou jsme vyvíjeli na cvičení.

První část úkolu se týká možné prohry. Aktuálně hra není nijak omezená – hráč může neustále zkoušet další a další pokusy, dokud číslo neuhodne a nevyhraje. V rámci tohoto úkolu byste měli do hry doplnit maximální počet pokusů pro uhodnutí čísla, po jehož vyčerpání hra automaticky skončí a hráč prohraje.

Druhá část úkolu se týká opakovaných tipů. Aktuálně si hra neuchovává žádné informace o tipovaných číslech. Pokud hráč provede platný, ale nesprávný tip, počet provedených pokusů se zvýší. Pokud hráč *(omylem)* tipne znovu stejné číslo, počet provedených pokusů se opět zvýší. V rámci tohoto úkolu byste měli toto chování změnit. Každé tipované číslo bude do počtu pokusů započítáno pouze jednou. Pokud hráč tipne číslo, které již tipnul dříve, hra na to upozorní zprávou: *„Toto číslo jsi již tipoval(a)“* a ke zvýšení počtu pokusů již nedojde.

Možnost prohrát

V archivu máte připravený BlueJ projekt s implementací hry. Oproti verzi ze cvičení je v něm upravená třída **`Game`**, která navíc obsahuje číselnou konstantu *(datový atribut označený modifikátory static a final)* **`MAX_GUESS_COUNT`** s hodnotou nastavenou na maximální povolený počet pokusů o uhodnutí čísla *(přednastaveno je 8 pokusů)*.

Projekt je třeba upravit následujícím způsobem:

- Naimplementovat metodu **`isLost()`**.
  Nyní metoda vždy vrací hodnotu **`false`**. Je třeba ji naimplementovat tak, aby vracela skutečný stav hry. Informaci o tom, zda hráč prohrál, je vhodné uchovávat v novém datovém atributu.
- Upravit implementaci ostatních metod.
  Určitě bude nutné upravit implementaci metody **`guess()`** tak, aby kontrolovala počet provedených pokusů a na základě toho nastavovala atributy indikující konec hry a výhru.
- Naimplementovat jednotkové testy.
  Třídu **`Game`** je potřeba otestovat automatizovanými testy. Můžete vyjít z testů, které jsme implementovali společně v rámci cvičení č. 4. Bude nutné upravit test metody **`isGameOver()`**, aby se v něm ověřilo, že hráč má skutečně k dispozici jen počet pokusů nastavený pomocí konstanty **`MAX_GUESS_COUNT`** a že po vyčerpání tohoto počtu pokusů hra skončí. Dále bude nutné doimplementovat test pro otestování metody **`isLost()`**.

Opakované tipy

Pro vyřešení opakovaných tipů je třeba projekt upravit následujícím způsobem:

- Přidat do třídy **`Game`** datový atribut pro ukládání již tipnutých čísel.
  Všechna tipnutá čísla je nutné nějakým způsobem ukládat, aby hra při vyhodnocování tipu mohla ověřit, zda dané číslo v minulosti již bylo tipováno, nebo ne. K uchovávání tipovaných čísel můžete použít např. pole logických hodnot *(datový atribut typu **`boolean[]`**)* o rozsahu 100 prvků. Každý prvek pole bude představovat informaci o tom, zda dané číslo *(index daného prvku)* již bylo tipnuto, nebo ne.
- Upravit implementaci metody **`guess()`**.
  Do metody je třeba doplnit kontrolu, zda je tipováno již tipnuté číslo. Pokud ano, metoda musí

vrátit text: *„Toto číslo jsi již tipoval(a)“*. Pokud ne, metoda musí uložit číslo do seznamu provedených tipů a provést vyhodnocení tipu.

- Upravit implementaci ostatních metod.
  Bude nutné upravit implementaci konstruktorů a metody **`startNewGame()`** tak, aby byl při zahájení nové hry seznam provedených tipů prázdný.

- Upravit a rozšířit testy.
  Bude nutné upravit test metody **`guess()`**, aby se v něm ověřilo, že v případě opakovaného tipu hráč dostane správnou zprávu. Dále bude nutné upravit test metody **`getGuessCount()`**, aby se v něm ověřilo, že opakované tipy se v průběhu hry nezapočítávají do celkového počtu pokusů a že po zahájení nové hry se seznam provedených tipů vymazal.

Třída **`Start`** je připravená tak, aby dokázala pracovat i s novou logikou hry. Pokud správně naimplementujete třídu **`Game`**, můžete spustit statickou metodu **`main()`** ve třídě **`Start`** a hru si zahrát a vyzkoušet.

Termín a hodnocení

- Je potřeba upravit implementaci hry a přidat automatizované testy.
- Termín pro splnění úkolu je **22. 11. 2020**.
- Za úkol můžete získat **6 bodů**.

Podklady

Podkladový projekt najdete v archivu se zadáním v adresáři **du-02-projekt**.

Odevzdání

Po vypracování úkolu ho odevzdejte do odevzdávárny **D. ú. 2 : Hádání čísel**.

Před odevzdáním vymažte v adresáři projektu všechny soubory s příponami **.ctxt** a **.class** a případnou vygenerovanou dokumentaci *(podadresář **doc**)*. Celý adresář s projektem poté zabalte do ZIP archivu.

Doplňte své jméno a datum do dokumentačních komentářů tříd, které jste v rámci úkolu vytvořili nebo upravili, viz následující příklad:

```
/**
 * Třída MojeTrida představuje ...
 *
 * @author: Antonín Kostěj
 * @version: 2020-11-22
 */
public class MojeTrida
{
    ...
}
```

Podrobnější poznámky k implementaci

Se správnou implementací první části tohoto úkolu se spousta studentů vždy poněkud ‚trápí‘. Zkusím proto doplnit ještě ‚epicky širší‘ popis fungování hry, který by Vám měl pomoci uvědomit si, co a jak je třeba ve hře upravit:

Třída **`Game`** obsahuje základní metody *(operace)*, které jsou nutné pro provoz hry. Jednak je to metoda **`guess`**, která provádí vyhodnocení hráčova tipu *(pokusu)*, dále jsou to metody **`getGuessCount`**, **`isGameOver`** a **`isLost`**, které informují o aktuálním stavu hry. Celý průběh hry *(komunikaci s uživatelem a volání výše uvedených metod ve správném pořadí)* zajišťuje třída **`Start`**. Hra v zásadě funguje následujícím způsobem:

- Volá se metoda **`isGameOver`**, pomocí které se zjistí, zda hra běží *(zda má smysl hádat)*.
- Pokud metoda **`isGameOver`** vrátí **`false`**:
  - Hra se zeptá uživatele na číslo, poté zavolá metodu **`guess`** a zadané číslo jí předá jako parametr.
  - Metoda **`guess`** zadané číslo vyhodnotí a její výsledek *(nepovolený tip / opakovaný tip / větší / menší / uhodl)* se vypíše na konzoli.
  - Poté se začíná od začátku *(voláním metody **`isGameOver`**, viz první bod)*.
- Pokud metoda **`isGameOver`** vrátí **`true`**:
  - Hra je ukončená a je potřeba uživateli sdělit výsledek.
  - Zavolá se metoda **`isLost`**, pomocí které se zjistí, zda hráč vyhrál, nebo prohrál.
  - Zavolá se metoda **`getGuessCount`**, pomocí které se zjistí, kolik tipů bylo provedeno.
  - Výsledky volání metod **`isLost`** a **`getGuessCount`** jsou použity k vypsání informací o tom, jak hra skončila.

Z výše uvedeného vyplývá několik věcí:

- Výsledek volání metody **`guess`** představuje jen a pouze vyhodnocení jednoho hráčova tipu, nenese žádnou informaci o stavu celé hry. Tzn. metoda **`guess`** bude vracet pouze texty:
  - **nepovolený tip**: Pokud hráč provedl tip mimo interval.
  - **opakovaný tip**: Pokud hráč znovu tipl již dříve tipované číslo
  - **větší**: Pokud hráč tipl příliš malé číslo.
  - **menší**: Pokud hráč tipl příliš velké číslo.
  - **uhodl**: Pokud hráč tipl správné číslo.
  - **hra neběží**: Tento výsledek se vrátí v případě, že by metoda byla zavolána ve chvíli, kdy nelze tipovat *(jedná se ale spíše o hypotetickou situaci, která by mohla nastat pouze v případě nějaké chyby – v našem případě se toto nikdy nestane, protože třída **`Start`** nebude volat metodu **`guess`** ve chvíli, kdy to nedává smysl)*. Místo této návratové hodnoty by bylo lepší z metody vyhodit výjimku, protože se jedná o situaci, ke které by za normálních okolností nemělo dojít. Výjimky budete na přednáškách probírat později.
- Nic jiného by metoda **`guess`** vracet neměla – tzn. žádné výsledky typu „prohrál jsi“, „neuhodl jsi ani na poslední pokus“ apod. Zjištění a vypsání těchto informací se provede zavoláním metod **`isGameOver`**, **`isLost`** a **`getGuessCount`**, metoda **`guess`** za toto ale zodpovědná není.
- Metoda **`isGameOver`** bude vracet informaci o tom, zda hra aktuálně běží. Tzn. na začátku hry bude metoda vracet hodnotu **`false`**. Jakmile hráč uhodne tajné číslo a nebo provede poslední povolený tah *(bez ohledu na to, zda uhodl, nebo ne)*, každé následné volání metody **`isGameOver`** již musí vrátit **`true`**.
- Metoda **`isLost`** bude vracet informaci o tom, zda hráč prohrál. Tzn. na začátku hry bude metoda vracet hodnotu **`false`**. Teprve pokud hráč vyčerpá povolený počet tahů a ani v posledním možném pokusu neuhodne tajné číslo, všechna následná volání metody **`isLost`** budou vracet **`true`**.

Na to, jakým způsobem a v jakém pořadí se volají metody třídy **`Game`**, se můžete podívat do kódu třídy **`Start`** *(řízení hry má na starost statická metoda **`play`** uvedená na konci třídy)*.