

Zadání samostatné práce z PPA1

Název: Jednoduchý generátor pseudonáhodných čísel, řazení a vizualizace posloupnosti náhodně generovaných prvků.

Vytvořte program, který bude **generovat pseudonáhodná čísla** podle následujícího algoritmu:

- 1) Program umocní zadané dvouciferné číslo (startovací hodnotu) na druhou.
- 2) Z výsledku použije první dvě cifry (zleva), čímž získá dvouciferné číslo.
- 3) K tomuto číslu přičte jedničku a výsledkem je nově vygenerované pseudonáhodné číslo.
- 4) Body 1) až 3) se opakují tak dlouho, dokud se nezačnou pseudonáhodná čísla opakovat (tj. končí výskytem pseudonáhodného čísla, které již bylo dříve vygenerováno).

Příklad vygenerování pseudonáhodných čísel:

$67 * 67 = 4489 \rightarrow 44 + 1 = 45$

$45 * 45 = 2025 \rightarrow 20 + 1 = 21$

$21 * 21 = 441 \rightarrow 44 + 1 = 45$ -- tato hodnota se již v posloupnosti vyskytla, generování končí

Posloupnost pseudonáhodných čísel začínající startovací hodnotou 67 je tedy 67, 45, 21

Řazení prvků vygenerované posloupnosti

Pro seřazení prvků takto vygenerované posloupnosti použijte jednu z metod řazení `SelectSort` (řazení výběrem), `InsertSort` (řazení vkládáním) nebo `BubbleSort` (řazení záměnou). Konkrétní metodu, kterou máte aplikovat, zjistíte na webové stránce v zadání cvičení, v němž je zadání SP vysvětlováno.

Vizualizace výsledků

Vytvořenou i vzestupně seřazenou posloupnost zobrazte použitím nástroje `DrawingTool`. Vzhled vizualizace, její rozměr, použité barvy, tloušťky čar (tj. sloupců) a rozložení obou posloupností je ponecháno na kreativě programátora s těmito omezeními:

- 1) Najednou musí být vidět celá původní i celá seřazená posloupnost.
- 2) Grafy obou posloupností budou obsahovat vodorovné úsečky, představující dělení osy Y na deset dílů. Díky tomu bude možno snadno ověřit správnost vizualizace.

Je možné se inspirovat vzhledem vizualizace dat vzorového příkladu: `pseudo10.jpg`

Ovládání programu

Program bude možné ovládat klávesnicí přímo z příkazové řádky s volbou spuštění programu bez nebo se zadáním parametru.

Ovládání se zadáním parametru

Bude-li na příkazové řádce jeden nebo více parametrů, budou tyto parametry představovat jednotlivé startovací hodnoty. Předpokládá se disciplinovaný uživatel, takže zadané hodnoty není třeba kontrolovat. Pro každou takovou startovací hodnotu vytvořte uvedeným způsobem pseudonáhodnou posloupnost. Tuto posloupnost zapište do první řádky výstupního textového souboru, který je nazván `vystup.txt`. Na následující řádku tohoto výstupního souboru zapište tutéž posloupnost, ovšem vzestupně seřazenou a oddělenou od dalšího výpisu prázdnou řádkou. Formát výpisu výsledků do souboru je závazný:

```
počet_prvků_posloupnosti [první_prvek, druhý_prvek, ... , poslední_prvek]
```

Například pro dvě startovací hodnoty 67 a 99 bude výpis do výstupního souboru:

Ukázka obsahu výstupního souboru:

```
3 [67, 45, 21]
3 [21, 45, 67]
```

```
1 [99]
1 [99]
```

Proveďte vizualizaci obou posloupností (tj. seřazené i neseřazené). Na obrazovku nic nevypisujte.

Ovládání bez zadání parametru

Nebude-li na příkazové řádce žádný parametr, program si nejprve vyžádá zadání jména vstupního souboru z klávesnice. Předpokládáme disciplinovaného uživatele, takže jméno souboru bude zadáno správně, tento soubor existuje a není třeba toto kontrolovat. Vstupní soubor bude textový a na každé řádce bude mít jedno dvouciferné číslo, které se bude považovat za startovací hodnotu. Počet řádek v souboru není omezen, jednotlivé startovací hodnoty se mohou opakovat, musí být však zadány správně (tj. dvouciferná čísla – není třeba kontrolovat).

Ukázka obsahu vstupního souboru:

```
10
11
83
```

Pro každou startovací hodnotu vytvořte již dříve uvedeným způsobem pseudonáhodnou posloupnost a vypište ji do jedné řádky na obrazovku. Na další řádku vypište tutéž posloupnost, ovšem seřazenou a oddělenou od dalšího výpisu prázdnou řádkou. Formát výpisu na obrazovku bude zcela totožný s již dříve uvedeným výpisem při spuštění programu s parametrem.

Například pro uvedený vstupní soubor bude výpis na obrazovku:

```
20 [10, 11, 13, 17, 29, 85, 73, 54, 30, 91, 83, 69, 48, 24, 58, 34, 12, 15, 23, 53]
20 [10, 11, 12, 13, 15, 17, 23, 24, 29, 30, 34, 48, 53, 54, 58, 69, 73, 83, 85, 91]

19 [11, 13, 17, 29, 85, 73, 54, 30, 91, 83, 69, 48, 24, 58, 34, 12, 15, 23, 53]
19 [11, 12, 13, 15, 17, 23, 24, 29, 30, 34, 48, 53, 54, 58, 69, 73, 83, 85, 91]

16 [83, 69, 48, 24, 58, 34, 12, 15, 23, 53, 29, 85, 73, 54, 30, 91]
16 [12, 15, 23, 24, 29, 30, 34, 48, 53, 54, 58, 69, 73, 83, 85, 91]
```

Po zpracování všech řádek ze vstupního souboru program ihned skončí. Nevykresluje tedy nic na obrazovku a ani nečeká na žádné další vstupy od uživatele.

Všechny výjimky řešte pouze pomocí `e.printStackTrace()`.

Další požadavky na program

Program vytvářejte důsledně strukturovaně s vytvořením a použitím minimálně pěti metod. Všechny metody i celou třídu důsledně okomentujte `JavaDoc` komentáři, ve kterých bude uveden `@author`. Nebude vyžadována žádná další dokumentace.

Odevzdání

Samostatnou práci odevzdejte **elektronicky** na portál **do bloku Samostatná práce** a to vložením zdrojového souboru s názvem **Ppa1_SP_OsobniCislo.java** (např. Ppa1_SP_A07999.java), který je závazný z důvodu snadné automatické kontroly (validace). Z důvodu validace je nezbytné **doplnit před výpisem výsledků na obrazovku** obdobně jako v domácích úlohách **příkaz**:

```
System.out.println("---Vysledky---");
```

Pokud pro vizualizaci nepoužijete nástroj `DrawingTool`, a došlo-li by k nějakému problému při validaci, pak celou vizualizaci před novým odevzdáním na portál zakomentujte. Pokud používáte Eclipse, pak rovněž před odevzdáním nezapomeňte odstranit `package`.

Obhájení

Samostatná práce bude **ohodnocena cvičícím (přidělení bodů) po jejím předvedení v době cvičení. Předpokladem pro předvedení je její předchozí úspěšná validace na portálu.**

Poznámky:

Na programu je třeba **začít pracovat co nejdříve** a podle vyložené látky vždy naprogramovat část. Doporučený postup je:

- 1) načtení startovací hodnoty z klávesnice
 - 2) algoritmus výpočtu dalšího čísla
 - 3) ukončení generování čísel
 - 4) vizualizace výsledků
 - 5) rozdělení programu do metod
 - 6) `JavaDoc` komentáře
 - 7) seřazení posloupnosti - lze po šesté přednášce
 - 8) dokončení programu víceméně pouze voláním již naprogramovaných metod a přesměrováním vstupů a výstupů ze/do souboru - lze po přednášce o souborech
- } - lze po třetí přednášce
} - lze po páté přednášce

To znamená, že v době zadání samostatné práce již máte dostatek vědomostí, aby bylo možné naprogramovat všechny klíčové metody a kompletně celé ovládání programu bez zadání parametru.