Semestrální práce

ALG1 2022/2023 – ÚLOHA 10

JIŘÍ ŠEPS

Zadání semestrální práce

Zapište program, který pro každé zadané kladné číslo (menší nebo rovné maximální hodnotě typu long jazyka Java) určí, zda číslo samo je zároveň zápisem své soupisky.

Číslo může být zadáváno jakékoliv v rozmezí hodnotě typu long jazyka Java, pokud zadané číslo bude záporné nebo 0, program se ukončí, číslo se píše bez mezer nebo jakéhokoliv jiného znaku

Návrh řešení

1) Rozdělní UI a metodu pro vstup a výpočet do jiných souborů
2) Jelikož pracujeme s čísly, které rozdělujeme na jednotlivé cifry, využijeme pole

    a) pole pro počet každých obsažených nenulových cifer v čísle

    b) pole pro ukládání jednotlivých cifer v metodě

3) Vytvoření metody která zjednodušeně spočítá kolik je jednotlivých cifer v zadaném čísle a přiřadí jim hodnotu.

Deklarace pole counts, a vytvoření instance new int [10], to nám umožní poté třídit čísla z pole deseti nul, 10, protože vybíráme z čísel 0-9, které se mohou vyskytnout v zadaném čísle

Využití num > 0, dělení intem zaokrouhluje dolů (9/10 = 0) tímto se zbavíme čísel která se v input čísle nevyskytují (aby to nevypsalo 01020304…)

Využití (% 10) modulo operátoru, který vlastně ‘‘izoluje‘‘ jednotlivé číslo z celkového input čísla, používáme 10, kvůli tomu aby to izolované číslo mohlo být pouze 0-9, protože větší zbytek mít nemůže, tento proces vlastně odstraní všechny čísla kromě číslice uplně v pravo, (10324 může být děleno 10 1032 krát se zbytkem 4, tímpádem izolovaná číslice je 4. Před modulo ještě je int, a to kvůli tomu abychom použili izolované číslo k indexaci pole, a díky tomu že to je int, dále použijeme / 10 abychom šli číslo po čísle protože int nemůže ukládat desetinná čísla (10324 /10 = 1032,4, ale kvůli intu se z toho stane 1032, které se poté dělí modulem znova) a poté je využito counts[digit]++, které indexuje číslo, jako třeba přiřadí hodnotě 4, jedničku.

Tato metoda je ztěžejní, protože počítá počet jednotlivých čísel v poli, jde to číslo po čísle, a podle počtu inkrementuje.

4) Vyvolání metody a vytvoření dalších dvou metod

    a) převedení do Stringu pro správný zápis výsledku, využít práce s polem

Deklarace Stringu result = ””, pro následné spojování počtu čísel, a jednotlivých čísel, dále klasické pracování s polem, a na konci result vznikné spojením counts[i] + “ + i, neboli počtu jednotlivého čísla, a číslo, kdyby tam nebyly “, tak by result číslo bylo součtem cifer a čísla, což nechceme, proto “

b) printování výsledku a napsání podmínky, zda se výsledek rovná zadanému číslu, nebo ne, použít if else.
Tato metoda je spíš jenom optimalizací, aby nemusela být v main class.

Testovací protokol

Číslo testu	Typ testu, popis vstupů	Očekávaný výsledek	Skutečný výsledek	Prošel (ano/ne)
1. 120651	Běžná hodnota	1021121516	1021121516	ano
2. -123	Nevalidní vstup	nic	nic	ano
3. 1111122222333334444455555666667777778888899999	Přes limitní stav	515253545556575859	error	ne
4. 1111111111112	Běžná hodnota	1112	1112	ano
5. 31123314	Běžná hodnota	31123314	31123314	ano

```
Zadej cislo:
120651
Cislo 120651 ma soupisku 1021121516
Cislo 120651 neni zapisem sve soupisky
```

```
Zadej cislo: -123
BUILD SUCCESSFUL (total time: 34 seconds)
```

```
Zadej cislo:
1111122222333334444455555666667777778888899999
Exception in thread "main" java.util.InputMismatchException: For input string: "1111122222333334444455555666667777778888899999"
        at java.base/java.util.Scanner.nextLong(Scanner.java:2379)
        at java.base/java.util.Scanner.nextLong(Scanner.java:2328)
        at Semestralka.Semestral.main(Semestral.java:19)
C:\Users\fiwie\AppData\Local\NetBeans\Cache\15\executor-snippets\run.xml:111: The following error occurred while executing this line:
C:\Users\fiwie\AppData\Local\NetBeans\Cache\15\executor-snippets\run.xml:68: Java returned: 1
BUILD FAILED (total time: 5 seconds)
```

```
Zadej cislo: 31123314
Cislo 31123314 ma soupisku 31123314
Cislo 31123314 je zapisem sve soupisky
```