**Lab02 - Výpočet statistiky číselné posloupnosti**

Template domácího úkolu [ke stažení zde](https://cw.fel.cvut.cz/wiki/_media/courses/b0b36pjv/hw/pjv-lab02.zip). Ke kontrole vstupů použijte funkce z třídy TextIO.

21.3.2018: Zamyslete se nad tím, že délka posloupnosti není omezená a není tedy vhodné vstupní data ukládat do paměti.

Úkoly:

Napište program, který vypočte průměrnou hodnotu a směrodatnou odchylku z posloupnosti čísel zadaných na standardní vstup. Při implementaci se inspirujte příkladem přesměrování standardního vstupu ze souboru. Testovací soubor obsahuje na každém řádku jedno číslo (nebo nečíselnou hodnotu, viz dále).

* Rozšiřte tento program pro výpočet statistiky (průměr a odchylka) z každých 10 vstupních čísel. Tyto dvě hodnoty vypište na jeden řádek standardního výstupu na tři desetinná místa a čísla oddělte mezerou, tj. formátování “%.3f %.3f”.
* Na začátek řádku vypište počet hodnot, ze kterých jsou průměr a odchylka vypočteny na dvě místa, tj. formátování “%2d”.
* Při detekci konce vstupního souboru vypište dílčí výsledek z příslušného počtu hodnot, ale pouze pokud je počet hodnot použitých k výpočtu vyšší než 1.
* Rozlište mezi výstupem na standardní výstup a na standardní chybový výstup. Program implementujte jako část třídy Lab02 v metodě pojmenované homework.
* Detekci konce vstupu indikujte výpisem 'End of input detected!' na standardní chybový výstup.

Odevzdávejte soubory Lab02.java, Start.java a TextIO.java.

V případě, že na řádku není detekováno číslo, program řádek přeskočí a pokračuje ve čtení vstupu do konce souboru. Tuto situace indikujte výpisem na standardní chybový výstup s uvedením čísla řádku, na kterém k této události došlo.

Pro výpočet směrodatné odchylky lze použít průměrnou hodnotu a průměr z mocniny sledované veličiny viz <http://en.wikipedia.org/wiki/Standard_deviation>.

Ukázka výstupu standardního a chybového výstupu po zpracování souboru [test\_input.txt](https://cw.fel.cvut.cz/wiki/_media/courses/b0b36pjv/hw/test_input.txt). Dbejte prosím na formátování výstupu: každý - i ten poslední - řádek by měl být zarovnaný správně a **ukončen znakem nového řádku** (viz. ukázka).

**java** -jar dist/lab02.jar <long\_input.txt

A number has not been parsed from line 6

10 53,500 23,851

10 56,000 26,446

10 34,400 19,184

10 53,900 26,121

10 63,400 32,116

10 58,200 28,071

10 65,800 32,508

10 44,200 24,891

A number has not been parsed from line 86

10 53,400 29,063

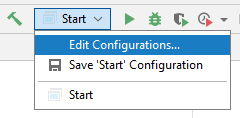
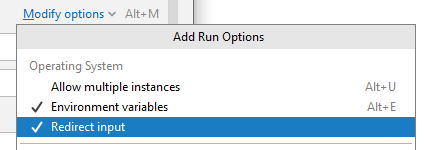
End of input detected!

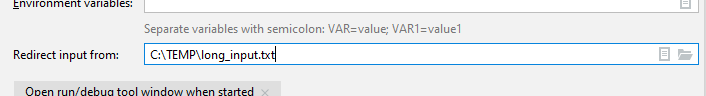
8 39,875 23,235

Tato ukázka využívá přesměrování textového suouboru do System.in, t.j. Váš program bude číst vstupy způsobenm, jako kdyby je zadával uživatel z klávesnice. Zajistí to znak přesměrování: <.

Alternativní způsoby přesměrování obsahu souboru na standardní vstup:

**IntelliJ Idea**



**Netbeans**

Pro účely testování přidejte do metody main kód:

public static void main([String](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+string)[] args) throws [FileNotFoundException](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+filenotfoundexception) {

[FileInputStream](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+fileinputstream) is = new [FileInputStream](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+fileinputstream)(new [File](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+file)("C:**\\**TEMP**\\**long\_input.txt"));

[System](http://www.google.com/search?hl=en&q=allinurl%3Adocs.oracle.com+javase+docs+api+system).setIn(is);

*//...*

Před odevzdáním řešení do BRUTE **tento kód pochopitelně odstraňte**!

Rada - spusteni jar souboru bez manifestu z prikazove radky:

java -cp ./soubor.jar cz.cvut.fel.pjv.Main