

09 - Generics, Sockets

Credits: Askar Gafurov

Dalsie materialy ku cviceniu:

- Vzorak ku casti Generics:
https://drive.google.com/file/d/1GbTepF4S_D_sxOMGT_NRjW5BJY4ZyTIX/view?usp=sharing
- Vzorak ku casti Sockets:
<https://drive.google.com/file/d/1yZuVyUtXGhNWSWMloYMyB2kstNDTT7nC/view?usp=sharing>

Generics - Wildcards

U: Vytvorte abstraktnú triedu Shape a konkrétnu triedu Circle s metódou draw().

U: Vytvorte dve polia: ArrayList<Shape> shapes, a ArrayList<Circle> circles, do oboch dajte tri objekty typu Circle.

U: Implementujte metódu drawAll(List<Shape> shapes) triedy Main. Viete pomocou nej vykresliť obe polia? Viete vstupné pole upravovať?

U: Implementujte metódu drawAll(List<?> shapes) triedy Main. Viete pomocou nej vykresliť obe polia? Viete vstupné pole upravovať?

U: Implementujte metódu drawAll(List<? extends Shape> shapes) triedy Main. Viete pomocou nej vykresliť obe polia? Viete vstupné pole upravovať?

Sockets

U: Spustíte príkaz ``nc -l 12345`` a v ďalšej konzole príkaz ``nc 127.0.0.1 12345``. Pohrajte sa s nimi.

U***: Skúste nadviazať spojenie s nejakým spolužiakom pomocou príkazu nc (lahke, keď ste v tej istej počítačovej sieti, pri rozdielnych sieťach si túto úlohu radšej odložte na neskôr).

U: Vytvorte triedu PortScanner s metódou scanRange(lower, upper), ktorá prejde všetky porty na počítači a s každým z nich sa pokúsi nadviazať spojenie.

U: Spustíte príkaz ``nc -l 12345`` pred spustením programu. Viete, čo sa stane po spustení metódy scanRange?

U: Upravte metódu scanRange() tak, aby vypisovala číslo aktuálne skúšaného portu zakaždým na ten istý riadok konzoly (symbol `"\r"`), a zachovával by na výstupe iba úspešné pokusy o spojenie.

U: Upravte metódu `scanRange()` tak, aby vedela oskenovať porty na ľubovoľnom hostname.

U: Skúste použiť metódu `scanRange` aj na hostname 'localhost', aj na Vašu IP adresu v rámci siete. Viete vysvetliť rozdiel?

U: Implementujte triedu `NetScanner` s metódou `scanRange(String prefix)`, ktorá má vypísať na konzolu, ktoré IP adresy s daným prefixom sú prítomné (rozlíšenie medzi `UnknownHostException` a `IOException`). Predpokladajte, že prefix je v tvare `XX.XX.XX`

U: Implementujte metódu `basic_communication()` triedy `Main`, ktorá sa pripojí ku portu 12345, prečíta si tri riadky zo vstupu a po každom načítaní vráti ten riadok so všetkými písmenami zmenenými do upper case.

U: Implementujte metódu `echo_server`, ktorá bude čakať na spojenie na porte 12345 (`ServerSocket serverSocket = new ServerSocket(...); Socket clientSocket = serverSocket.accept()`). Po nadviazaní spojenia bude donekonečna vracať načítaný vstup so zmenenými písmenkami do upper case.

U: Zabaľte dané volanie do nekonečného while-loopu. Je to korektný spôsob, ako zabezpečiť postupné spracovanie všetkých klientov daným echo serverom?

U: Implementujte korektným spôsobom echo server, ktorý po ukončení spojenia s jedným klientom sa bude venovať ďalšiemu klientovi.