Notatki - Scanner

Spis treści

Scanner

Cały czas rozmawialiśmy o zapisywaniu danych do plików. Wspomnieliśmy też sobie w międzyczasie o `System.in`, czyli Streamie, który pozwala na odczyt danych wprowadzanych przez użytkownika.

Od razu też uprzedzę, że w pracy komercyjnej jest mała szansa, że ta klasa będzie Ci potrzebna, ale warto wiedzieć że coś takiego jest. Czasem może się przydać w testowaniu.

Powiedzieliśmy, że aby wydrukować na ekranie informacje podczas działania aplikacji mamy dostępne 2 `PrintStreamy`: `System.out` i `System.err`. Oprócz nich jest jeszcze `InputStream` - `System.in`. `System.in` jest "surowym" `InputStreamem`. Możemy natomiast uprościć jego używanie stosując klasę `java.util.Scanner`, która opakowuje Stream `System.in` w taki sposób, żebyśmy mogli odwoływać się do całych wprowadzanych w terminalu linijek tekstu.

Przykład wykorzystania klasy `Scanner`:

```
public class ScannerExample {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter some data");

        while (scanner.hasNext()) {
            String line = scanner.nextLine();
            System.out.println("Entered: " + line);

            if ("done".equals(line)) {
                break;
            }
        }
    }
}
```

W przykładzie powyżej możemy wprowadzać dane do programu do momentu aż wpiszemy `done`.

Scanner vs BufferedReader

Znamy już na tym etapie możliwość czytania danych za pomocą klas "...Reader" oraz za pomocą klasy `Scanner`. Jaka jest zatem różnica? Kiedy używać której z nich?

Gdy zaczniemy o tym czytać w internecie to natkniemy się na takie rozróżnienie, że `Scanner` służy do

parsowania, a "...Reader" do czytania. No dobrze, to jaka jest różnica między jednym, a drugim.

- parsowanie - odczyt danych z jakiegoś miejsca z jednoczesną próbą zrozumienia co te dane oznaczają w jakimś kontekście. Czyli czytamy próbując jednocześnie wyjąć z tych danych pewne informacje, które mogą być dla nas znaczące. Prostym przykładem jest fragment tekstu reprezentujący zestaw tagów HTML. Z jednej strony jest to zwykły tekst, ale parsując go możemy przedstawić ten sam tekst jako ustrukturyzowany zestaw tagów HTML.
- czytanie - jak sama nazwa mówi, program może czytać dane bez jednoczesnej próby interpretacji ich znaczenia.

Jak się to ma to klas `Scanner` oraz np. `BufferedReader`? Klasa `Scanner` daje nam metody takie jak np. `nextInt()` lub `nextBigDecimal()`, które świadczą o tym, że przy odczycie próbujemy od razu konwertować te dane na jakiś kontekst, który jet dla nas zrozumiały. Dla odmiany `BufferedReader` nie ma takich metod, służy on tylko do czytania danych, bez próby ich rozumienia w jakimś kontekście.