

Zadania z programowania w języku Java dla II roku Informatyki

Na podstawie książki: „Java 9. Przewodnik doświadczonego programisty”

Interfejsy i wyrażenia lambda

1. Utwórz interfejs `Measurable` z metodą `double getMeasure()`, która dostarcza jakąś metrykę obiektu. Zaimplementuj interfejs `Measurable` w klasie `Employee`. Utwórz metodę **`double average(Measurable[] objects)`**, która oblicza średnią metryk. Wykorzystaj ją do obliczenia średniego wynagrodzenia pracowników, których dane są zapisane w tablicy.
2. Kontynuując poprzednie ćwiczenie, utwórz metodę **`Measurable largest(Measurable[] objects)`**. Wykorzystaj ją do ustalenia nazwiska pracownika z najwyższym wynagrodzeniem. Do czego zostanie użyte rzutowanie?
3. Jaki jest typ nadrzędny dla typu `String`? Dla typu `Scanner`? Dla typu `ImageOutputStream`? Zauważmy, że każdy typ jest swoim własnym nadtypem. Klasa lub interfejs bez zadeklarowanego typu nadrzędnego otrzymuje jako typ nadrzędny klasę `Object`.
4. (a) W klasie `IntSequenceClass`, która implementuje interfejs `IntSequence` ze slajdu nr 21 zaimplementuj statyczną metodę **`public static of(int...)`**, która zwraca ciąg argumentów. Przykładowo, wywołanie `IntSequenceClass.of(3, 1, 4, 1, 5, 9)` zwraca instancję klasy `IntSequenceClass` przechowującą ciąg sześciu wartości. Kolejne wywołania metody `next` zwracają kolejne liczby przechowywane w tej instancji.
(b) Zaimplementuj statyczną metodę **`public static of(int...)`** w interfejsie `IntSequence`, która zwraca instancję wewnętrznej klasy anonimowej. Kolejne wywołania metody `next` zwracają kolejne liczby przechowywane w tej instancji.
5. Zaimplementuj metodę statyczną `constant` w klasie `IntSequence`, która zwraca nieskończony ciąg stałych. Na przykład `IntSequence.constant(1)` zwraca wartości `1 1 1 ...`, w nieskończoność.
6. Klasa `SquareSequence` nie dostarcza w rzeczywistości nieskończonej sekwencji kwadratów z powodu przepełnienia całkowitego. Jak konkretnie się zachowuje? Napraw problem poprzez zdefiniowanie interfejsu `Sequence<T>` oraz klasy `SquareSequence`, która implementuje klasę `Sequence<BigInteger>`.