

Zadania z programowania do przedmiotu „Zaawansowane programowanie w Javie”

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

Wykład 2

Interfejsy i wyrażenia lambda

7. W tym ćwiczeniu masz za zadanie sprawdzić, co się stanie po dodaniu metody do interfejsu. W Java 7 zaimplementuj klasę `DigitSequence`, która implementuje `Iterator<Integer>`, nie `IntSequence`. Utwórz metody `hasNext`, `next` i nierobiącą niczego metodę `remove`. Napisz program, który wyświetla elementy instancji. W Java 8 klasa `Iterator` ma inną metodę, `forEachRemaining`. Czy Twój kod nadal się kompiluje po przejściu na Java 8? Jeśli umieścisz klasę z Java 7 w pliku JAR i nie będziesz jej kompilować ponownie, czy zadziała w Java 8? A co, jeśli wywołasz metodę `forEachRemaining`? Poza tym metoda `remove` stała się domyślną metodą w Java 8 wywołującą wyjątek `UnsupportedOperationException`. Co się stanie, jeśli metoda `remove` zostanie wywołana na instancji Twojej klasy?
8. Zaimplementuj metodę

```
void luckySort(ArrayList<String> strings, Comparator<String> comp),
```

która wywołuje `Collections.shuffle` na tablicy typu `ArrayList` do chwili, gdy elementy będą uporządkowane rosnąco w sposób określony przez komparator.
9. Zaimplementuj klasę `Greeter`, która implementuje interfejs `Runnable` i w której metoda `run` wyświetla `n` kopii tekstu `"Witaj, " + target`, gdzie `n` i `target` są ustawiane w konstruktorze. Stwórz dwie instancje z różnymi komunikatami i wykonaj je równolegle w dwóch wątkach.
10. Zaimplementuj metody:

```
public static void runTogether(Runnable... tasks)
public static void runInOrder(Runnable... tasks)
```

Pierwsza metoda powinna uruchomić każde zadanie w oddzielnym wątku i zakończyć działanie. Druga metoda powinna uruchomić wszystkie zadania w bieżącym wątku i zakończyć działanie po zakończeniu ostatniego z nich.