

# Ćwiczenie: stabilność sortowania

Napisać program, który potwierdza stabilność metody `sort()` z klasy `Arrays` i niestabilność metody sortowania `quicksort()` podanej w programie `Std03_SortowanieTablicowe`.

1. Utworzyć nowy projekt o nazwie **SortStability**.
2. Skopiować metodę `quicksort()` z **Std2\_3\_SortowanieTablicowe** i zmodyfikować ją do postaci generycznej z relacją porządku zadawaną przy pomocy dodatkowego parametru typu `Comparator<E>`.
3. Skopiować do projektu klasę `Point` (np. z **Std2\_7\_GenericArrayHeap**).
4. Zorganizować w metodzie `main()` zwykłą konwersację dla losowej generacji tablicy  $n$  punktów `Point[] P`; generowane punkty mają postać  $(x, i)$ , gdzie współrzędna  $x$  pochodzi z generatora losowego, a współrzędna  $y$  jest numerem punktu,  $i = 0, 1, \dots, n-1$ .
5. Pokazać wygenerowane dane i efekt sortowania przy pomocy `Arrays.sort(P, c)` oraz `quicksort(P, c)`, gdzie  $c$  jest komparatorem zakodowanym przy pomocy wyrażenia `lambda` porządkującego rosnąco wg współrzędnej  $x$  punktów.

Stabilność sortowania będzie potwierdzona, jeśli punkty o tych samych wartościach współrzędnych  $x$  będą uporządkowane rosnąco wg współrzędnej  $y$  (czyli numeru porządkowego generacji). Wg tego samego schematu postępowania można zbadać stabilność innych metod sortowania.