

Programowanie Java
Anna Gogolińska
Zestaw

We wszystkich zadaniach niżej używaj rachunku lambda zamiast pętli typu *for each*.

Zad 1. Używając rachunku lambda napisz program, w którym będzie wypisywana zawartość katalogu podanego jako argument, ale tylko pliki tekstowe większe niż 10KB.

Zad 2. Napisz klasę *Person*, z polami imię, nazwisko, wiek, e-mail oraz odpowiednimi metodami setter, getter, konstruktor. Następnie stwórz listę obiektów klasy *Person*. Posortuj ją kilka razy, za każdym razem używając sortowania „po” innym polu (po imieniu, po nazwisku, po wieku). Za każdym sortowaniem wypisz zawartość listy na ekran. Do sortowania proszę użyć interfejsu *Comparator* oraz rachunku lambda. Dla ułatwienia można najpierw napisać ten program bez użycia rachunku lambda.

Zad 3. Utwórz klasę wątku implementując interfejs *Runnable*. Wątek ma losować liczbę naturalną n od 0 do 100, a następnie wypisać na ekran wszystkie liczby parzyste między 0, a n wraz z liczbą n . Użyj rachunku lambda. Utwórz trzy wątki na podstawie klasy zdefiniowanej z *Runnable* i uruchom je.

Zad 4. Stwórz klasę *Dog* (można używać też polskich nazw), która będzie zawierała różne pola (rasa, kolor, waga, wysokość, imię itd.). Do tego odpowiednie metody. Następnie stwórz interfejs *CatalogueDog*, która będzie zawierała jedną metodę *checkBelong(Dog d)*, która będzie zwracać *boolean*. Następnie stwórz kilka/kilkanaście obiektów klasy *Dog* i dodaj je do listy. Później używając klas implementujących *CatalogueDog* (mogą być anonimowe) na podstawie listy obiektów klasy *Dog* stwórz kilka list psów o wspólnych cechach, jak np. duże, małe, czarne itd. Napisz to zadanie bez wyrażeń lambda, a następnie z wyrażeniami lambda i użyciem gotowego interfejsu funkcyjnego.

Zadania samodzielne

Zad 5. Do zadania 4 dopisz interfejs, z metodą *String print(Dog d)*. Użyj klas implementujących ten interfejs (mogą być anonimowe) do wypisywania w różny sposób informacji o obiekcie podanym jako argument, np. tylko części informacji, lub inaczej sformatowanych. W drugiej wersji rozwiązania użyj rachunku lambda i gotowego interfejsu funkcyjnego (można pominąć rozwiązanie bez rachunku lambda).