

Laboratorium z programowania obiektowego

20.05.2009

1. Napisz program, który wypisuje na ekran zawartość pliku tekstowego. Nazwę pliku do wypisania wczytaj ze standardowego wejścia za pomocą odpowiednich strumieni.
2. Napisz program, który dla podanego pliku wypisze, z ilu bajtów składa się ten plik.
3. Napisz program, który dla danego pliku tekstowego wypisze liczbę znaków oraz liczbę białych znaków w tym pliku.
4. Napisz program, który kopiuje znak po znaku pliki tekstowe.
5. Zmień program z poprzedniego zadania tak, aby czas kopiowania był mierzony i wypisywany na konsolę. Sprawdź, jaki wpływ na efektywność ma użycie buforowania.
6. Napisz `Iterator<String>`, w którym metoda `next()` zwraca kolejne linie wczytane z `Reader`a. Konstruktor ma dostawać obiekt `Reader`a. Po wczytaniu wszystkiego `Reader` ma zostać zamknięty. Zauważ, że `Iterator` nie może rzucać wyjątków `IOException`. Niech przechwyci taki wyjątek i pozwoli normalnie iterować po wszystkich poprawnie wczytanych liniach.
Wskazówka: `BufferedReader` ma metodę `readLine()`; zwraca ona `null`, gdy napotka koniec pliku.
Wskazówka 2: aby `hasNext()` działał poprawnie, trzeba zawsze mieć wczytaną i spamiętaną jedną linię w przód.
7. Przy wypisywaniu zawartości katalogu domyślne sortowanie jest leksykograficzne. Zatem plik `laboratorium12.txt` jest wypisywany przed plikiem `laboratorium2.txt`. Napisz własny komparator, który sortuje tak, że nazwa `laboratorium2.txt` jest wcześniej niż nazwa `laboratorium12.txt`. Użyj go do wypisania w odpowiedniej kolejności listy plików z wybranego katalogu.

Wskazówka. Do odczytywania zawartości katalogu można użyć klasy `File`.