Dany jest pakiet napisy z definicją interfejsu Napis , implementującej go klasy Zwykły oraz klasy Fabryka .

- Metoda zwykły() klasy Fabryka daje napis o określonej treści.
- Metoda doklej() skleja dwa napisy.
- Metoda powtórz() powtarza napis określoną liczbę razy.
- Metoda fragment() daje fragment napisu, zaczynający się od pozycji określonej pierwszym argumentem, mający długość określoną drugim argumentem.

  Zakładamy, że drugi argument nie przekracza różnicy między długością napisu i wartością pierwszego argumentu.
- Metoda długość() daje długość napisu.
- Metoda toString(), przedefiniowana dla napisu, daje jego treść.

Uzupełnij pakiet o alternatywną, "efektywną" reprezentację napisów, spełniającą poniższe warunki.

- Reprezentacja efektywna gwarantuje wykonanie metod doklej(), powtórz(), fragment() i długość() w czasie stałym.
- Metoda toString() dla reprezentacji efektywnej działa w czasie liniowym względem sumy długości napisu i liczby obiektów, z których jest zbudowany.
- W klasie Fabryka jest dodatkowa, działająca w czasie stałym, metoda:

```
public static Napis efektywny(String string) {
    ...
}
```

która daje efektywną reprezentację napisu o treści przekazanej jako argument.

- W pakiecie napisy jest dodatkowa klasa TestNapisów z metodą main(), która demonstruje użycie napisów.
- Wskazówki:

  - Definicja reprezentacji efektywnej powinna się składać hierarchii klas, po jednej dla każdego sposobu utworzenia napisu.
  - Liniowość metody toString() można osiągnąć za pomocą odpowiednich definicji metody appendTo().