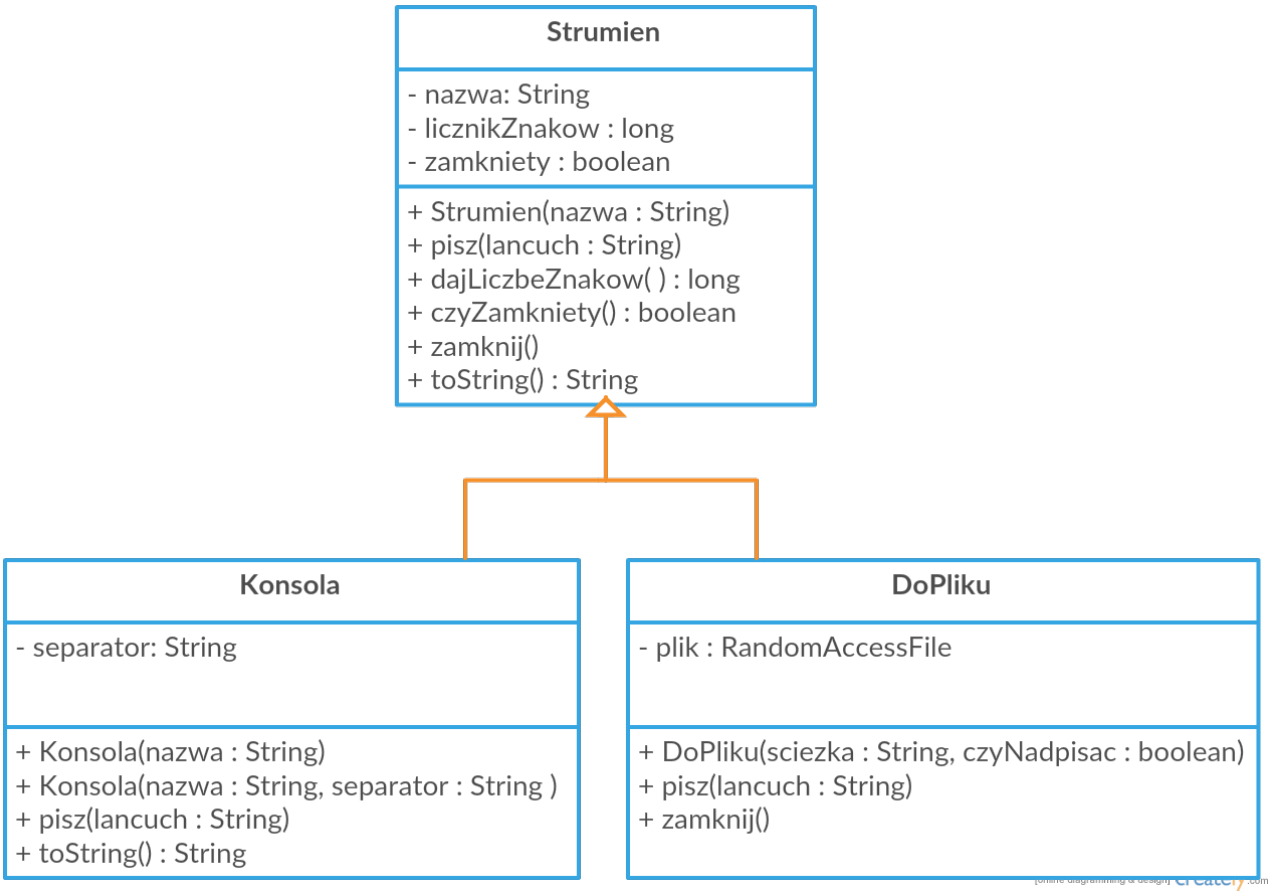
Zadanie Strumienie

Część I

Zaimplementuj klasy pokazane na diagramie.



Klasy te umożliwiają zapisywanie łańcuchów do *strumieni*. Dla każdej należy zdefiniować odpowiednie konstruktory i wskazane metody. Pola oznaczone symbolem minus „-” powinny być prywatne, natomiast oznaczone symbolem plus „+” powinny być publiczne. Działanie każdej z metod należy zweryfikować.

Klasa `Strumien` jest klasą *bazową*, przy czym:

- pole `nazwa` pełni rolę informacyjną i ustalane jest przy konstruowaniu strumienia;
- metoda `pisz` zwiększa wartość pola `licznikZnakow` o liczbę wypisanych do strumienia znaków, poza tym metoda ta nie wykonuje żadnych operacji;
- po wykonaniu metody `zamknij` pole `zamkniety` otrzymuje wartość `true`, co oznacza, że do strumienia nie można już więcej zapisać żadnych łańcuchów;
- metoda `toString` wyświetla nazwę strumienia, liczbę wypisanych znaków oraz informacje o tym, czy strumień jest zamknięty.

Klasa `Konsola` wypisuje łańcuchy na standardowe wyjście, dodatkowo:

- pole `separator` służy do oddzielania kolejnych napisów przekazywanych do strumienia,

tzn. wyświetlane jest na zakończenie każdego wywołania metody `pisz;`

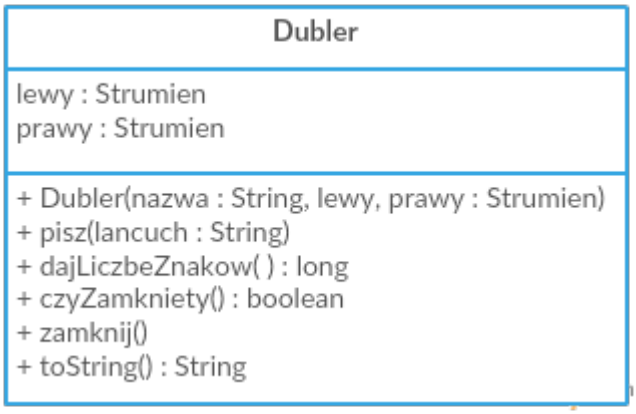
- konstruktor pierwszy ustala wartość pola `separator` na „\n”;
- konstruktor drugi otrzymuje dodatkowo parametr `separator` zdefiniowany przez użytkownika.

Klasa `DoPliku` wypisuje łańcuchy do pliku za pomocą klasy `RandomAccessFile:`

- ścieżka do pliku przekazywana jest w parametrze konstruktora, ścieżka ta staje się automatycznie nazwą strumienia z klasy bazowej (`Strumien`);
- plik otwierany jest automatycznie w konstruktorze, w przypadku niepowodzenia powinien zostać wyświetlony komunikat o błędzie, a metoda `pisz` nie powinna próbować zapisywać niczego do pliku;
- metoda `zamknij` nadpisuje metodę z klasy bazowej i dodatkowo zamyka plik.

Część II

Zaimplementuj klasę `Dubler`.



Klasa `Dubler` również dziedziczy z klasy `Strumien`, jednak:

- zamiast wypisywać łańcuchy do nowego strumienia, wypisuje wszystkie przekazywane łańcuchy do dwóch istniejących strumieni: `lewy` i `prawy` przekazanych w konstruktorze (może to być ten sam strumień dwukrotnie);
- istnieje możliwość, że oba przekazane strumienie nie będą istnieć, tj. `null`, wszystkie metody powinny to sprawdzać i nie wykonywać operacji na nieistniejącym strumieniu składowym;
- metoda `dajLiczbeZnakow` powinna zwracać sumę znaków zapisanych do obu strumieni;
- metoda `czyZamkniety` powinna zwracać `true`, jeżeli oba strumienie są zamknięte;
- metoda `zamknij` powinna zamykać oba strumienie;
- metoda `toString` powinna oprócz nazwy strumienia wyświetlić informację o obu strumieniach składowych oraz łącznej liczbie wypisanych znaków i stanie strumienia.