**Klasy abstrakcyjne:**

- nie mogą być inicjowane słowem new, mogą być tylko dziedziczone przez inne klasy i w ten sposób użyte

- jeśli klasa posiada co najmniej jedną metodę abstrakcyjną to klasa też musi być abstrakcyjna

- stosować gdy potrzeba zaimplementować jakieś wspólne działanie dla wszystkich klas dziedziczących

- klasy dziedziczące muszą implementować metody abstrakcyjne

- ponieważ dziedziczenie jest ograniczone tylko do jednej klasy to czasem lepiej jest stosować interfejsy.

**String konstruktor vs nadanie wartości:**

Ogólnie jest co najmniej jedna sytuacja kiedy lepiej użyć konstruktora Stringa zamiast zwykłej implementacji. Kiedy tworzymy nowego stringa który jest substringiem poprzedniego. Następująca sytuacja:

String s = "0123456789012345678901234567890123456789";

String s2 = s.substring(0, 1);

s = null;

W tej sytuacji string ‘s’ nie zostanie usunięty przez garbage collector ponieważ tak deklarowane stringi nigdy nie są usuwane przez VM. Żeby pierwszy string został usunięty należy użyć konstruktora i napisać kod w następujący sposób.:

String s2 = new String(s.substring(0, 1));

W ten sposób string pierwszy zostanie usunięty i powstanie nowy string o nazwie s2.

**Final na zmiennej:**

Dla typów prymitywnych oznacza że raz nadana wartość nie może być zmieniona. Dla typów obiektowych oznacza tyle że adres przechowywyany przez zmienną nie może się zmienić natomiast sam obiekt już tak (np. można zmieniać wartości pól takiego obiektu).

**Final na klasach:**

Oznacza że klasa nie może być dziedziczona (przykładowe klasy to String oraz System). Metody w klasie final automatycznie też są final.