

---

---

# KURS JĘZYKA JAVA

## ZBIORY ASOCJACYJNE

### Zadanie 1.

W pakiecie `struktury` definiuj klasę `Para`, która będzie przechowywać pary klucz–wartość, gdzie klucz jest identyfikatorem typu `String` a skojarzona z nim wartość to liczba rzeczywista typu `double`. Klucz powinien być publicznym polem niemodyfikowalnym (powinien składać się tylko z małych liter alfabetu angielskiego oraz nie może to być łańcuch pusty ani wartość `null`), a wartość polem prywatnym, które można odczytać za pomocą gettera i zmodyfikować tylko za pomocą settera.

```
public class Para {
    public final String klucz;
    private double wartość;
    // ...
}
```

W klasie tej zdefiniuj metody `toString()` oraz `equals(Object)` — dwie pary są równe, gdy mają takie same klucze.

### Zadanie 2.

W pakiecie `struktury` zdefiniuj abstrakcyjną klasę `Zbior`, która będzie przechowywać różne pary (w zbiorze nie mogą wystąpić dwie pary o takim samym kluczu).

```
public abstract class Zbior {
    /** metoda ma wyszukać parę z zadaniem kluczem
     * metoda zgłasza wyjątek, gdy nie znajdzie pary o podanym kluczu */
    public abstract Para szukaj (String k) throws Exception;
    /** metoda ma wstawić do zbioru nową parę
     * metoda zgłasza wyjątek, gdy para o podanym kluczu już jest w zbiorze */
    public abstract void wstaw (Para p) throws Exception;
    /** metoda ma usunąć ze zbioru parę o zadaniem kluczu
     * gdy pary o podanym kluczu nie ma w zbiorze metoda nic nie zrobi */
    public abstract void usuń (String k);
    /** metoda ma odszukać parę o zadaniem kluczu i zwrócić wartość pozwiązaną
     * metoda zgłasza wyjątek, gdy nie znajdzie pary o podanym kluczu */
    public abstract double czytaj (String k) throws Exception;
```

```

    /** metoda ma wstawić do zbioru nową albo zaktualizować parę */
    public abstract void ustaw (Para p) throws Exception;
    /** metoda ma usunąć wszystkie pary ze zbioru */
    public abstract void czysc ();
    /** metoda ma podać ile jest wszystkich par w zbiorze */
    public abstract int ile ();
}

```

Klasa ta ma być klasą bazową dla klas imlementujących zbiór.

### Zadanie 3.

W pakiecie `struktury` zdefiniuj klasę `ZbiorNaTablicy` dziedziczącą po klasie `Zbior` implementującą zbiór na zwykłej tablicy par. Rozmiar tablicy podaj do konstruktora poprzez argument (wartość ta nie może być mniejsza od 2). W przypadku jak zabraknie miejsca na wstawienie nowej pary, to należy zgłosić wyjątek.

### Zadanie 4.

W pakiecie `struktury` zdefiniuj klasę `ZbiorNaTablicyDynamicznej` dziedziczącą po klasie `ZbiorNaTablicy` implementującą zbiór na tablicy dynamicznej. Początkowy rozmiar tablicy niech wynosi 2. W przypadku jak zabraknie miejsca na nową parę, to tablicę należy powiększyć dwukrotnie (utorzyć nową tablicę dwa razy większą i przepisać do niej dane ze starej tablicy). Jeśli przy usuwaniu elementów zapelnienie tablicy spadnie poniżej  $\frac{1}{4}$  jej pojemności, to zmniejsz tablicę dwukrotnie.

### Zadanie 5.

Uzupełnij swoje zadanie o krótki program testowy napisany w pakiecie `test`. Program ma rzetelnie sprawdzić działanie obiektów typu `ZbiorNaTablicy` i `ZbiorNaTablicyDynamicznej` reprezentujących zbiory.

### Uwaga.

Implementując klasy dla zbiorów asocjacyjnych nie korzystaj z żadnej kolekcji standardowej.