

LABORATORIUM

ZAAWANSOWANE PROGRAMOWANIE OBIEKTOWE

Stream API

1. Wstęp

Wstęp Ćwiczenie ma na celu zapoznanie z Stream API.

2. Przebieg ćwiczenia

Utworzyć projekt maven lub gradle.

- Przekleić zawartość załączników do plików Ingredients.java i Pizza.java, uzupełnić brakujące importy. Każda Pizza posiada nazwę (name) i listę składników (ingredients). Każdy składnik posiada nazwę (name), cenę (price), informację czy jest to mięso (meat) (np pizza wegetariańska nie może posiadać żadnego składnika mięsnego), informację czy jest pikantny (spicy). Cena pizzy to suma cen wszystkich jej składników.
- Zaimplementować klasę z następującymi metodami:
 - Pizza findCheapestSpicy() - metoda zwracająca najtańszą ostrą pizzę.
 - Pizza findMostExpensiveVegetarian() - metoda zwracająca najdroższą pizzę wegetariańską.
 - List iLikeMeat() - metoda zwracająca same pizzę mięsne, posortowane malejąco po liczbie składników mięsnych.
 - Map groupByPrice() - metoda grupujące pizzę po cenie.
 - String formattedMenu() - metoda zwracająca string w postaci nazwa_pizzy: składnik1, składnik2, składnik3 - cena, kolejne pizzę oddzielone znakiem nowej linii.
- Wszystkie metody zaimplementować z wykorzystaniem Stream API, najlepiej w postaci pojedynczego wyrażenia. Wewnątrz metod nie mogą się znajdować instrukcje sterujące (warunki, pętle).
- DODATKOWE: Utworzyć testy jednostkowe dla wszystkich zaimplementowanych metod.

Załączniki

Ingredients:

```
public enum Ingredient {  
    THIN_CRUST("cienkie ciasto", 12, false, false),  
    THICK_CRUST("grube ciasto", 14, false, false),  
    TOMATO_SUACE("sos pomidorowy", 0, false, false),  
    CHEESE("ser", 1, false, false),  
    ONION("cebula", 1, false, false),  
    PINEAPPLE("ananas", 1, false, false),  
    BROCCOLI("brokuły", 1, false, false),  
    BEAN("fasola", 1, false, false),  
    CORN("kukurydza", 1, false, false),  
    PEPPERONI("papryka peperoni", 1, false, true),  
    MUSHROOMS("pieczarki", 1, false, false),  
    TOMATO("pomidor", 1, false, false),  
    FETA("ser feta", 2, false, false),  
    MOZARELLA("mozzarella", 2, false, false),  
    SUASAGE("kiełbasa", 2, true, false),  
    CHICKEN("kurczak", 2, true, false),  
    OLIVES("oliwki", 2, false, false),  
    SALAMI("salami", 2, true, false),  
    TUNA("tuńczyk", 2, false, false),  
    BECON("bekon", 3, true, false),  
    ARUGULA("rukola", 3, false, false),  
    BLUE_CHEESE("ser pleśniowy", 2, false, false),  
    HAM("szynka", 3, true, false),  
    BASIL("bazylia", 0, false, false),  
    PEPPER("pieprz", 0, false, true),  
    TABASCO_SUACE("tabasco", 0, false, true) ;  
    private final String name;  
    private final int price;
```

```

    private final boolean meat;

    private final boolean spicy;

    private Ingredient(String name, int price, boolean meat,
boolean spicy) {

        this.name = name;

        this.price = price;

        this.meat = meat;

        this.spicy = spicy;

    }

    /**
     * Zwraca nazwe skladnika
     * @return
     */
    public String getName() {
        return name;
    }

    /**
     * Zwraca cene skladnika
     */
    public int getPrice() {
        return price;
    }

    /**
     * zwraca true jesli skladnik jest miesem (nie dla
    vegetarian)
     */
    public boolean isMeat() {
        return meat;
    }

    /**
     * zwraca true jesli skladnik jest pikantny

```

```

*/

public boolean isSpicy() {
    return spicy;
}

@Override
public String toString() {
    return name;
}
}

```

Pizza:

```

public enum Pizza {

    MARGHERITA("Marrgherita", THIN_CRUST, TOMATO_SUACE,
CHEESE),

    CAPRI("Capri", THIN_CRUST, TOMATO_SUACE, CHEESE, HAM,
MUSHROOMS),

    HAVAI("Havai", THIN_CRUST, TOMATO_SUACE, CHEESE,
HAM,PINEAPPLE),

    CARUSO("Caruso", THIN_CRUST, TOMATO_SUACE, SUASAGE,
PEPERONI),

    MAMA_MIA("Mama Mia", THIN_CRUST, TOMATO_SUACE, CHEESE,
ONION, MUSHROOMS, BECON),

    SOPRANO("Soprano", THICK_CRUST, TOMATO_SUACE, CHEESE, HAM,
MUSHROOMS, ONION, BECON, PEPPER),

    CALABRESE("Calabrese", THICK_CRUST, TOMATO_SUACE, CHEESE,
HAM, MUSHROOMS, SUASAGE, ONION, OLIVES),

    VEGETARIANA("Vegetariana", THIN_CRUST, TOMATO_SUACE,
CHEESE,

    ONION, BEAN, CORN, BROCCOLI, ARUGULA),

    CAPRESE("Caprese", THICK_CRUST, TOMATO_SUACE,
MOZARELLA,FETA, TOMATO, BASIL),

    PASCETORE("Pascetore", THIN_CRUST, TOMATO_SUACE, CHEESE,
TUNA, ONION),

    FOUR_CHEESE("Cztery sery", THIN_CRUST, TOMATO_SUACE,
CHEESE,MOZARELLA, FETA, BLUE_CHEESE),

```

```

        TABASCO("Tabasco", THICK_CRUST, TOMATO_SUACE, CHEESE, HAM,
SALAMI, PEPERONI, CORN, TABASCO_SUACE),

        AMORE("Amore", THIN_CRUST, TOMATO_SUACE, CHEESE,
CHICKEN, TOMATO),

        FARMER("Farmerska", THICK_CRUST, TOMATO_SUACE, CHEESE,
CHICKEN, BECON, ONION, CORN);

private final String name;
private final List<Ingredient> ingredients;
private Pizza(String name, Ingredient... ingredients) {
    this.name = name;
    this.ingredients = Arrays.asList(ingredients);
}
public String getName() {
    return name;
}
public List<Ingredient> getIngredients() {
    return ingredients;
}
@Override
public String toString() {
    return name;
}
}

```