

OOP Intro

DEADLINE: 29/01/2022

FOLDER STRUCTURE



TASK

Task 1:

You can create pizza of different sizes and different types:

Sizes:

- S (+50 UAH)
- M (+75 UAH)
- L (+100 UAH)

Types:

- VEGGIE (+50 UAH)
- MARGHERITA (+60 UAH)
- PEPPERONI (+70 UAH)

You can add extra ingredients to the pizza:

Extra ingredients:

- CHEESE (+7 UAH)
- TOMATOES (+5 UAH)
- MEAT (+9 UAH)

Write a program that calculates the cost of a pizza and info about your pizza. Use an OOP approach (**hint: you need a *Pizza* class, constants, methods for choosing options and calculating the required values**).

The code must be error-proof. Imagine another programmer using your class. If it passes the wrong type of pizza, for example, or the wrong kind of ingredient, an exception should be thrown (the error should not be silently ignored) (**hint: you need *PizzaException* class**).

Example of errors:

```
let pizza = new Pizza(Pizza.SIZE_S);
```

```
✖ ▶ Uncaught ▶ PizzaException {log: "Required two arguments, given: 1"}
```

```
let pizza = new Pizza(Pizza.SIZE_S, Pizza.SIZE_S);
```

```
✖ ▶ Uncaught ▶ PizzaException {log: "Invalid type"}
```

```
let pizza = new Pizza(Pizza.SIZE_S, Pizza.TYPE_VEGGIE);
```



Frontend Lab

```
pizza.addExtraIngredient(Pizza.EXTRA_MEAT);
pizza.addExtraIngredient(Pizza.EXTRA_MEAT);
```

✖ ▶ Uncaught ▶ PizzaException {log: "Duplicate ingredient"}

```
let pizza = new Pizza(Pizza.SIZE_S, Pizza.TYPE_VEGGIE);
pizza.addExtraIngredient(Pizza.EXTRA_CORN);
```

✖ ▶ Uncaught ▶ PizzaException {log: "Invalid ingredient"}

Pizza Class Description

Class members:

- properties (size and type are required):

- *size*: size of pizza (must be from the allowedSizes property)
- *type*: type of pizza (must be from the allowedTypes property)

- methods:

- *addExtraIngredient(ingredient)*: add extra ingredient. Method must:
 1. Accept only one parameter, otherwise show an error
 2. Check if such an ingredient exists in allowedIngredients, if does not exist, show an error
 3. Check if such an ingredient already exists; if there is, show error (you can add one ingredient only once)
- *removeExtraIngredient(ingredient)*: remove extra ingredient. Method must:
 1. Accept only one parameter, otherwise show an error
 2. Check if such an ingredient exists in allowedIngredients, if does not exist, show an error
 3. Check if such an ingredient has already been added, if it not added, show an error, otherwise remove ingredient
- *getSize()*: returns size of pizza
- *getPrice()*: returns total price
- *getPizzaInfo()*: returns size, type, extra ingredients and price of pizza

PizzaException Class Description

Provides information about an error while working with a Pizza. Details are stored in the log property.

Class members:

- properties:

- *log*: information about an error while working with a Pizza.

RESTRICTIONS

- Using any external libraries are forbidden.
- Adding **task/** folder is forbidden. Do not push it to repository. (Only **homework/** folder should be pushed)

BEFORE SUBMIT

- Code should be clean, readable, and tested
- Make sure your GitLab folder structure meets folder structure from this document (without **task** folder)
- Use linter :
 - In order to use npm package manager you should install nodejs (<https://nodejs.org/>)



- Install eslint to check your code (npm install -g eslint)
- open a terminal (or cmd)
- run eslint (i.e. eslint ./js/index.js)

Code should be without 'errors'

SUBMIT

- The **FL18_HW3** folder without **task** folder should be uploaded to GitLab repository "**FL-18**" into **main** branch.