CO2 emission calculator

Your task is to create a program that returns the amount of CO2-equivalent that will be caused when traveling between two cities using a given transportation method.

Organizational

This task has a number of functional and non-functional requirements that are listed below. You should not work longer than 8 hours on this task. Make sure you prioritize the most important requirements first. Please submit your solution even if you were not able to finish everything within 8 hours.

The task should be completed using Java or NodeJS. Make sure to upload all required files to our submission system. Avoid uploading unnecessary files like binaries or dependencies, but please do include a `README.md` that describes how to install dependencies, compile and execute the solution.

If you have questions regarding this task, feel free to send an email and we will get back to you as soon as possible.

CO2 data

For the calculation, please use the following average values.
Transportation methods in CO2e per passenger per km:

- Small cars:
  - `small-diesel-car`: 142g
  - `small-petrol-car`: 154g
  - `small-plugin-hybrid-car`: 73g
  - `small-electric-car`: 50g
- Medium cars:
  - `medium-diesel-car`: 171g
  - `medium-petrol-car`: 192g
  - `medium-plugin-hybrid-car`: 110g
  - `medium-electric-car`: 58g
- Large cars:
  - `large-diesel-car`: 209g
  - `large-petrol-car`: 282g
  - `large-plugin-hybrid-car`: 126g
  - `large-electric-car`: 73g
- `bus`: 27g
- `train`: 6g
  Source: BEIS/Defra Greenhouse Gas Conversion Factors 2019

Geocode and distance API

Use the openrouteservice to get the distance between 2 cities. Please create a free account to get an API Token. If you have problems getting a token, get in contact with us. You don't need to share your token with us. Please read the value of the token from an environment variable called `ORS_TOKEN`.

To solve the task you can use the following endpoints:

- https://openrouteservice.org/dev/#/api-docs/geocode/search/get: Search for a city by name to get the coordinates
  - Provide parameters `api_key` and `text` (the city name)
  - Optionally provide parameter `layers="locality"` (to limit the search to cities)
  - It returns a list of matching locations ordered by confidence
- https://openrouteservice.org/dev/#/api-docs/v2/matrix/{profile}/post : Get the time or distance between two cities
  - In the body, provide `locations` (list of coordinates) and `metrics=["distance"]`
  - Provide URL parameter `profile=driving-car`
  - Provide header `Authorization=API_KEY`
  - It returns a matrix of distances for the given locations

Using these APIs are suggestions, feel free to adjust to your needs.

Acceptance Criteria

Functional requirements:

- The tool can be called with two cities `start` and `end` as well as a `transportation-method`. It outputs the amount of CO2-equivalent in kilogram.

```
$ ./co2-calculator --start Hamburg --end Berlin --transportation-method medium-diesel-car
Your trip caused 49.2kg of CO2-equivalent.
```

- Named parameters can be put in any order and either use a space ( ) or equal sign (`=`) between key and value.

```
$ ./co2-calculator --start "Los Angeles" --end "New York" --transportation-method=medium-diesel-car
Your trip caused 770.4kg of CO2-equivalent.

$ ./co2-calculator --end "New York" --start "Los Angeles" --transportation-method=large-electric-car
Your trip caused 328.9kg of CO2-equivalent.
```

Non-Functional requirements:

- The implemented features are **unit tested**
- Errors and edge-cases are considered
- The implementation uses a dependency management tool which allows easy compilation and test execution (e.g. in an CI/CD environment).
- The `README.md` file contains clear instructions on how to compile, test and execute the tool.

- Compilation is possible with Windows, Linux and macOS
- The API token is stored and read from an environment variable called `ORS_TOKEN`
- Best practices regarding architecture and code style are considered.