

## Wariant 171

- Utwórz pakiet `sports.performanceIndex` do obliczania indeksu wydajności sportowej.
- Zdefiniuj interfejs `PerformanceIndexCalculator` z metodami: `calculateIndex(int score)` zwracającą `double` (obliczającą indeks wydajności na podstawie wyniku) i `isValidScore(int score)` zwracającą `boolean` (sprawdzającą, czy wynik jest prawidłowy).
- Stwórz klasę `SimplePerformanceIndexCalculator`, która implementuje `PerformanceIndexCalculator`. Metoda `calculateIndex` powinna obliczać indeks jako  $\text{Math.sqrt}(\text{score}) * 1.5$ . Metoda `isValidScore` powinna sprawdzać, czy `score` jest większe lub równe 0.
- W klasie `TestPerformanceIndexCalculator` utwórz 2 obiekty `SimplePerformanceIndexCalculator` i przetestuj metody.

Zadanie należy umieścić we własnym repozytorium na Githubie.

## Wariant 172

- Utwórz pakiet `sports.speedCalculator` do obliczania prędkości sportowców.
- Zdefiniuj interfejs `SpeedCalculator` z metodami: `calculateSpeed(double distance, double time)` zwracającą `double` (obliczającą prędkość) i `isValidDistance(double distance)` zwracającą `boolean` (sprawdzającą, czy dystans jest dodatni).
- Stwórz klasę `SimpleSpeedCalculator`, która implementuje `SpeedCalculator`. Metoda `calculateSpeed` powinna obliczać prędkość jako  $\text{distance} / \text{time}$ . Metoda `isValidDistance` powinna sprawdzać, czy `distance` jest większe od 0.
- W klasie `TestSpeedCalculator` utwórz 2 obiekty `SimpleSpeedCalculator` i przetestuj metody.

Zadanie należy umieścić we własnym repozytorium na Githubie.

## Wariant 173

- Utwórz pakiet `sports.bodyMassIndex` do obliczania wskaźnika masy ciała (BMI).
- Zdefiniuj interfejs `BodyMassIndexCalculator` z metodami: `calculateBMI(double weight, double height)` zwracającą `double` (obliczającą BMI) i `isValidHeight(double height)` zwracającą `boolean` (sprawdzającą, czy wzrost jest prawidłowy).
- Stwórz klasę `SimpleBodyMassIndexCalculator`, która implementuje `BodyMassIndexCalculator`. Metoda `calculateBMI` powinna obliczać BMI jako  $\text{weight} / (\text{height} * \text{height})$ . Metoda `isValidHeight` powinna sprawdzać, czy `height` jest większe niż 0.
- W klasie `TestBodyMassIndexCalculator` utwórz 2 obiekty `SimpleBodyMassIndexCalculator` i przetestuj metody.

Zadanie należy umieścić we własnym repozytorium na Githubie.

## Wariant 174

- Utwórz pakiet `travel.fuelCostEstimator` do szacowania kosztów paliwa dla podróży.
- Zdefiniuj interfejs `FuelCostCalculator` z metodami: `calculateCost(double distance, double fuelEfficiency, double fuelPrice)` zwracającą `double` (obliczającą koszt paliwa na podstawie odległości, efektywności zużycia paliwa i ceny paliwa) i `isPositiveNumber(double number)` zwracającą `boolean` (sprawdzającą, czy liczba jest dodatnia).
- Stwórz klasę `SimpleFuelCostCalculator`, która implementuje `FuelCostCalculator`. Metoda `calculateCost` powinna obliczać koszt jako  $\text{distance} / \text{fuelEfficiency} * \text{fuelPrice}$ . Metoda `isPositiveNumber` powinna sprawdzać, czy `number` jest większy od 0.
- W klasie `TestFuelCostCalculator` utwórz 2 obiekty `SimpleFuelCostCalculator` i przetestuj metody.

Zadanie należy umieścić we własnym repozytorium na Githubie.