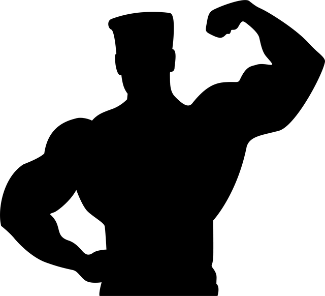
**03. Workout**

****

*Help your friend Peter to get fit. Write a program that creates exercises and workouts.*

**Preparation**

Download the skeleton provided in Judge. **Do not** change the **packages**!

**Pay attention to the name of the package workout, all the classes, their fields, and methods the same way they are presented in the following document. It is also important to keep the project structure as described.**

**Problem description**

Your task is to create a workout, which stores exercises by creating the classes described below.

**Exercise**

First, write a Java class **Exercise** with the following properties:

* **name: String**
* **muscle: String**
* **burnedCalories: int**

The class **constructor** should receive **name, muscle,** and **burnedCalories**. You need to create the appropriate **getters and setters**. Override the **ToString()** method in the following format:

**"Exercise: {name}, {muscle}, {burnedCalories}"**

**Workout**

**Next**, write a Java class **Workout** that has **exercises** (**List**, which stores the entity **Exercise**). All entities inside the repository have the **same fields**. Also, the Workout class should have those fields:

* **type: String**
* **exerciseCount: int**

The class **constructor** should receive **type** and **exerciseCount**, also it should initialize the **exercises** with a new instance of the list**.** Implement the following features:

* Field **exercises** – **List** that holds added exercises.
* Method **addExercise(Exercise exercise)** – **adds** an entity to the exercises If there is still **space on** the workout sheet **(exerciseCount).**
* Method removeExercise(String name, String muscle) – removes the exercise by **given name and muscle,** if such **exists**, and **returns boolean**.
* Method **getExercise(String name, String muscle)** – returns the exercise with the given **name** and **muscle** or **null** if there is no such exercise.
* Method getMostBurnedCaloriesExercise() – returns the exercise which is burned the most calories or null if there are no exercises.
* Getter getExerciseCount() – **returns** the **number** of exercises.
* **getStatistics()** – **returns** a **String** in the following **format**:
  + **"Workout type: {workout type}  
    Exercise: {Exercise1}  
    Exercise: {Exercise2}  
    (…)**"

**Constraints**

* The **combinations** of **names** and **muscles** will be **always unique**.
* The **burned calories** from the exercises will always be **positive**.
* There won't be exercises with the same burned calories.

**Examples**

This is an example of how the **Workout** class is **intended to be used**.

|  |
| --- |
| **Sample code usage** |
| // Initialize the repository  Workout workout = newWorkout("strength", 3);  // Initialize entity  Exercise exercise = newExercise("Bench Press", "chest", 30);  // Print Exercise  System.out.println(exercise); // Exercise: Bench Press, chest, 30  // Add Exercise  workout.addExercise(exercise);  // Remove Exercise  System.out.println(workout.removeExercise("Bench Press", "arms")); // falseSystem.out.println(workout.removeExercise("Bench Press", "chest")); // true  //Get exerciseSystem.out.println(workout.getExercise("Bench Press", "chest")); // null  Exercise secondExercise = newExercise("Deadlift", "back", 50); Exercise thirdExercise = newExercise("Barbell Curl", "biceps", 25); Exercise fourthExercise = newExercise("Squats", "legs", 60); Exercise fifthExercise = newExercise("Deadlift", "legs", 55);  workout.addExercise(secondExercise); workout.addExercise(thirdExercise); workout.addExercise(fourthExercise); workout.addExercise(fifthExercise);  //Get mostBurnedCaloriesExercise  Exercise mostBurnedCaloriesExercise = workout.getMostBurnedCaloriesExercise(); System.out.println(mostBurnedCaloriesExercise); // Exercise: Squats, legs, 60  Exercise exerciseByNameAndMuscle = workout.getExercise("Deadlift", "back"); System.out.println(exerciseByNameAndMuscle); // Exercise: Deadlift, back, 50  // Count  System.out.println(workout.getExerciseCount()); // 3System.out.println(workout.removeExercise("Barbell Curl", "biceps")); // true  // Print Statistics  System.out.println(workout.getStatistics());  //Workout type: strength //Exercise: Deadlift, back, 50 //Exercise: Squats, legs, 60 |

**Submission**

Submit a **single .zip file**, containing the **workout package, with the classes inside (Exercise, Workout, and the Main class**, there is no specific content required inside the Main class e. g. you can do any kind of local testing of your program there. However, there should be a **main(String[] args)** method inside.