Simple Fitness Tracker

Objectives

- Getting Started with JAVA Programming Language.
- Becoming familiar with Java IDE (e.g., eclipse, NetBeans, ...).
- Develop a simple GUI to input and display application data.
- Apply OOP principles to your GUI code.

Description

It is required to implement simple fitness tracking system with the following rules:

- User has four types of activities: SWIMMING, RUNNING, KICK_BOXING and STRENGTH_TRAINING.
- Each of those four activities has an effect on **calories burnt** and **heart rate** as shown in the following table:

ACTIVITY	CALORIES BURNT/minute	HEART RATE INCREASE/minute
SWIMMING	4	0.2%
RUNNING	5	0.3%
KICK_BOXING	3	0.5%
STRENGTH_TRAINING	5	0.6%

Task

- You can assume initially calories burnt are 0 and heart rate is 80 beat/minute.
- You should allow the user to enter the type of activity and time taken to practice (in minutes).
- User can enter the same type of activity many times.
- After the user confirms the activity, a success message should appear (e.g "Activity Recorded Successfully").
- Each time the user enters an activity you should calculate calories burnt and heart rate increase according to activity type and time using the table above, for simplicity you can assume the following:
    - time is an integer number.
    - heart rate after performing an activity is calculated using the following formula:

$$new\ heart\ rate\ =\ old\ heart\ rate + (old\ heart\ rate * time\ taken * heart\ rate\ increase)$$

- You should allow the user to get total calories burnt and total heart rate after doing some activities.
- You should also allow the user to sort activities according to their contribution in burning calories in descending order. If two activities have the same impact on calories burnt, sort them according to their contribution in heart rate increase in descending order as well.
- You should create at least **2 classes**: one for logic and calculations and one for GUI.
- You are free to use any GUI components (buttons, drop down list, labels, text boxes, ...etc) to achieve the goal.
- You should at least have two windows: one for entering the activity and time taken and one for getting results (total calories burnt and heart rate **or** activities ranking).
- You can use drag and drop tools for this assignment (e.g. window builder, ...etc), but you are encouraged not to use them.

Sample Run

Add activity:
Type: SWIMMING, **Time**: 5 minutes.
(total calories burnt = 0+5*4=20, total heart rate = 80+80*5*0.002=80.8)

Add activity:
Type: KICK_BOXING, **Time**: 8 minutes.
(total calories burnt = 20+8*3=44, total heart rate = 80.8+80.8*8*0.005=84.032)

Add activity:
Type: SWIMMING, **Time**: 4 minutes.
(total calories burnt = 44+4*4=60, total heart rate = 84.032+84.032*4*0.002=84.704)

Add activity:
Type: RUNNING, **Time**: 9 minutes.
(total calories burnt = 60+9*5=105, total heart rate = 84.704+84.704*9*0.003=86.991)

Add activity:
Type: STRENGTH_TRAINING, **Time**: 5 minutes.
(total calories burnt = 105+5*5=130, total heart rate = 86.991+86.991*5*0.006=89.601)

Results:

total calories burnt: 130 calories.
total heart rate: ~89.601 beat/minute.

Activities Rank:

1. RUNNING:
   calories burnt: 45 calories
   heart rate increase: 2.287 beat/minute.

2. SWIMMING:
   calories burnt: 36 calories.
   heart rate increase: 1.472 beat/minute.

3. STRENGTH_TRAINING:
   calories burnt: 25 calories
   heart rate increase: 2.609 beat/minute.

4. KICK_BOXING:
   calories burnt: 24 calories
   heart rate increase: 3.232 beat/minute.

Deliverables

- Develop this assignment in Java programming language.
- You should submit your code online at the deadline, wait for the submission instructions on Piazza.
- You can use any graphics libraries (e.g. Swing, SWT, FX, ...).
- You should work individually.
- [**Cheating Policy**] Delivering a copy will be severely penalized for both parties, so delivering nothing is so much better than delivering a copy.

Useful Resources:

- JavaFX basics: JavaFX
- JavaSwing: JavaSwing
- Google is your best friend ☺.