Kristianstad University
SE-291 88 Kristianstad
Sweden
+46 44 20 30 00
www.hkr.se

Homework C# Basics: Module 3

Note:

- ***Each task has its own project***
- ***The projects will be added to the same solution (see the earlier assignment document "inlämningsuppgifter2")***
- ***Exception handling/validation of input value shall be considered in all of the tasks when needed/appropriate. (It will affect your grade!)***
- ***You have the freedom to develop the exercises further so that it reflects something exciting from real world scenarios!***

Homework Assignments

1. Create a program and add a new class to it with the name Sphere.

    a. The Sphere class shall have one private field of the type integer that represents the radius of the sphere.

    b. Write a default, empty, constructor and also a constructor that takes one parameter, the sphere's radius.

    c. Write a property that allows the class user to modify the radius. Validate the value so that the negative values for radius shall set the radius to zero.

    d. Write two method that calculate the volume and surface area of the sphere.

    e. Test your class from the Main-method. To test the class you can create one or more objects of the class (in the Main-method), and modify the radius both with the constructor and with the property and print the modified value on the console window. Call your method and calculate and print the volume and surface area of the sphere. Do the test with at least two different values for radius. You have the freedom to choose whether you want to ask the user for input values or you want to test it with some pre-defined, hard-coded values.

2. A Simplistic Bank application

In this task you, shall create a small, attractive and robust bank application. Use the example code in Canvas (Module 3) and also lecture 4. Use also your creativity to make the application as user friendly and robust as possible.

Your application shall have a menu that allows the user to choose between different options. An example of menu can be:

Kristianstad University
SE-291 88 Kristianstad
Sweden
+46 44 20 30 00
www.hkr.se


```
Enter 1 to deposit money, 2 to withdraw money and 3 to leave.
1
You want to put some money in, enter the amount!
```

or

```
Enter 1 to deposit money, 2 to withdraw money and 3 to leave.
2
You want to take out some money in, enter the amount!
```

or

```
Enter 1 to deposit money, 2 to withdraw money and 3 to leave.
3
By by!
```

or

```
Enter 1 to deposit money, 2 to withdraw money and 3 to leave.
45
What do you means with this number?
```

You can use the example above or you can create your own menu which can be much better, may be with more options.

Add needed methods like ***withdraw*** to your Account class. Remember that you need to validate the input values. For example if the amount of money that user wants to withdraw is more than the current balance, the application shall print some text message like *"Sorry, you don't have enough money in your account"*.

Minimum requirements for this task:

- A welcome message containing name of the user (the application shall ask user's name before printing the message!) and date and time.
- A menu that allows the user to **continue choosing different options** until the user choose to leave/terminate the application.
- Deposit, withdraw and leave/terminate functions.
- Error handling and validating of input values!

  You can of course develop your application by adding more options, creating a nicer menu and so on, which will affect your grade.

3. (A Simplistic Bicycle Class)

Your task is to write a class **Bicycle**, according to the following description:
Suppose you want to create a number of Bicycle objects of that class and assign each of them, a serial number (Id), beginning with 1001 for the first object.
Description of the class:

- Your class holds pieces of information including Id, speed of the Bicycle (an integer) and model (can be a string).
- The value of the instance variable, speed will be initialized to zero in the constructor of your class. Initiate the value of model to your favourite model.
- Provide a property to read and modify the speed.
- Provide a read-only property to **only read the model** (no set block).

Kristianstad University
SE-291 88 Kristianstad
Sweden
+46 44 20 30 00
www.hkr.se

- Provide two methods, ***Accelerate*** and ***Brake***. The Accelerate method adds 5 to speed and the Brake method subtract 5 from the speed.
- **Note**: If the speed is over 100, it should not be possible to increase tit any further. The speed should not be below zero under any conditions either.

Test your class in an application. (As explained in task 1e, create some objects of your class call different methods, and print all the information (including Id-number) on the console window, or simply provide a menu for the user to choose between different options (menu is not a requirement!)).

(Hint: Declare a static variable in your class like Bicycle_Id, and increase id by one in your constructor. In that way you can keep track of number of objects that has been created, and you know what ID to assign to the next object)

4. (Target-Heart-Rate Calculator)

While exercising, you can use a heart-rate monitor to see that your heart rate stays within a safe range suggested by your trainers and doctors. For more on this, readthe note from the American Heart Association (AHA) (www.heart.org/HEARTORG/GettingHealthy/PhysicalActivity/Target-Heart-Rates_UCM_434341_Article.jsp)

The formula for calculating your **maximum heart rate** in beats per minute is **220 minus your age** in years. Your **target heart rate** is a range that is **50–85% of your maximum heart rate**. [Note: These formulas are estimates provided by the AHA. Maximum and target heart rates may vary based on the health, fitness and gender of the individual. Always consult a physician or qualified health care professional before beginning or modifying an exercise program.]

Create a class called ***HeartRates***. The class attributes should include the person's first name, last name, year of birth and the current year. Your class should have a constructor that receives this data as parameters.

For each attribute provide a property with set and get accessors. The class also should include a method that calculates and returns the person's age (in years), a method that calculates and returns the person's maximum heart rate and methods that calculate and return the person's minimum and maximum target heart rates. Write an app that prompts for the person's information, instantiates an object of class HeartRates and displays the information from that object—including the person's first name, last name and year of birth—then calculates and displays the person's age in (years), maximum heart rate and target-heart-rate range.

Requirements for grades:
Grade 3: Tasks 1, 2 solved.
Grade 4: 3 tasks solved (Tasks 1 and 2, plus one more task).
Grade 5: All the 4 tasks solved.

Your code should be neat and readable. Before uploading your solutions, make sure that you have done validation of the input values and exception handling in all tasks.

If you have questions, mail me at:

dawit.megistu@hkr.se

(Credits to Nazila)

Good Luck!