

Practice

Clean code

Read: <http://www.c-sharpcorner.com/article/important-tips-to-write-clean-code-in-visual-studio/>,
<https://www.codeproject.com/Articles/539179/Some-practices-to-write-better-Csharp-NET-code>

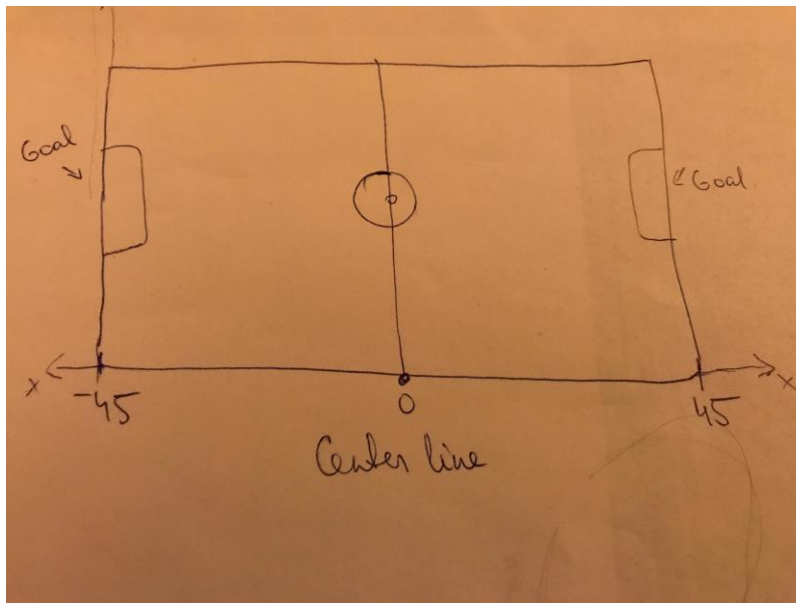
Take the bad code example from here and try to fix it, DO NOT peek the solution before you've tried yourself.

<https://www.codeproject.com/Articles/1083348/Csharp-BAD-PRACTICES-Learn-how-to-make-a-good-code>

Exercise:

In every area of life there is more and more technology used. Soccer is no difference. You're hired by FIFA to help them with creating a prototype for „digital football“.

Since this is a testing prototype, the scenarios are simplified. Ball movement is measured only from x axis coordinates, where 0 is the value of center line. Right edge of the field is 45; left edge is -45. Y axis is not taken into consideration.



The ball has:

- Method for calculating and returning the average speed for the ball. This method takes starting position, final position and time as parameters and calculates the average speed based on them. Formula is: $v=s/t$

The positions are given as x-axis values. The positions can be also negative so take this into consideration. The result has to always be positive.

For example:

CalculateAverageSpeed(30, 70, 20) returns $((70-30)/20)= 2$.

CalculateAverageSpeed(30, -10, 20) returns $((-10-30)/20)= 2$.

CalculateAverageSpeed(30, 0, 20) returns $((0-30)/20)= 1.5$.

- Method to determine whether the goal counted or not based on the coordinates of the ball (coordinates for x axis are the method parameter). To count as a goal, the whole ball has to be inside the goal (not just a part of it).
To simplify, we assume that the goal is shaped like a box. The coordinates represent the middle point of the ball. The ball is inside a goal if:
 $\text{ballCoordinates} - \text{ballRadius} \geq \text{goalCoordinates} - \text{goalDepth}$
- Method to calculate the kinetic energy of hitting the ball with foot. This method takes velocity as a parameter and returns kinetic energy. Kinetic energy is calculated by:
 $E = mv^2/2$, where m is the weight of the ball and v is the velocity.
- Method to return the total number of goals that counted and the number of goals that did not count.
- Method for creating and printing out an unique code for the ball.
 - For the normal ball the code is generated from four first letters from the sponsor name and then 5 random numbers.
 - For the youth ball the code is generated from three first letters from the sponsor name and 3 random numbers.
 - Think of the possible exceptions and avoid them or catch them!

There are two different balls:

- Normal ball diameter is 70 cm. With normal ball the goal depth is always 1.7m. The weight of the ball is 450g.
- Youth ball diameter is 56 cm. With this ball the goal depth is 1.4m. The weight is 380g.
- Both balls have constructors that take sponsor name as a parameter.

In main method create both balls. With both of them:

- Try checking if the ball was inside the goal with 20 random coordinate values (test also negative ones). Try also the situations where the coordinates are invalid. After doing that print out how many goals were in and how many out. The sum of them should be 20.
- Calculate average speed for 10 random coordinates.
- Find kinetic energy for velocity values ranging from 1 to 5.
- Create an unique code for both of the balls.

Tips:

How to solve:

1. Write down all the methods and properties the classes should have and find which class should be the base class.
2. Implement base class and test it in main method.

3. If base class works, then start creating derived classes.
- Take into consideration the limits for the coordinates; coordinates can range from -45 to 45. If any other value is inserted, a custom „CoordinateNotValid“ exception should be thrown. This exception should be also caught with try-catch block. Tip: make a separate method to check the validity of the coordinates. How to create custom exceptions: <https://stackify.com/csharp-exception-handling-best-practices/>
 - Start by thinking through what classes should have what methods and how they differ. Start writing from the base class.