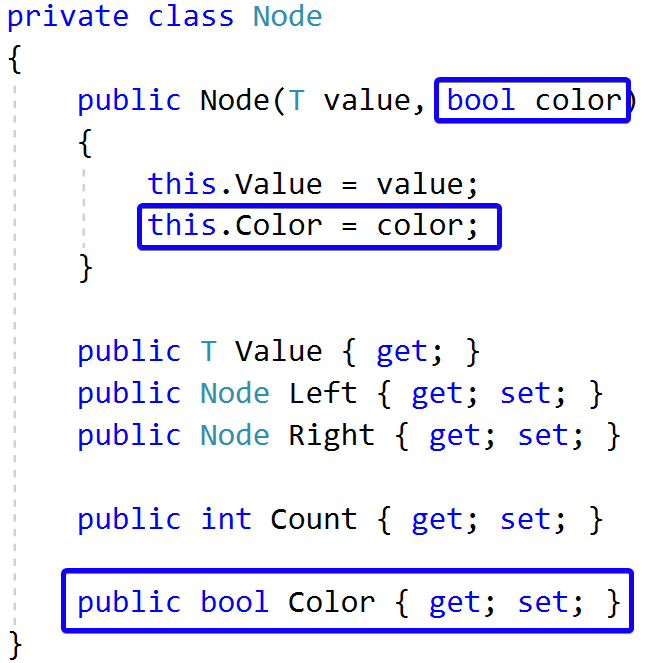
# Lab: Red-Black Trees and AA-Trees Lab

This document defines the lab for ["Data Structures – Advanced (C#)" course @ Software University](https://softuni.bg/trainings/3113/data-structures-advanced-with-csharp-october-2020).

Please submit your solutions (**source code**) of all below described problems in [Judge](https://judge.softuni.bg/Contests/2603/03-Red-Black-Trees-and-AA-Trees-Lab).

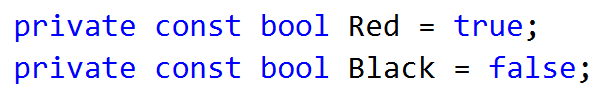
## Change the Node Data Structure

First, you will need to add a **color bit** to our node class:

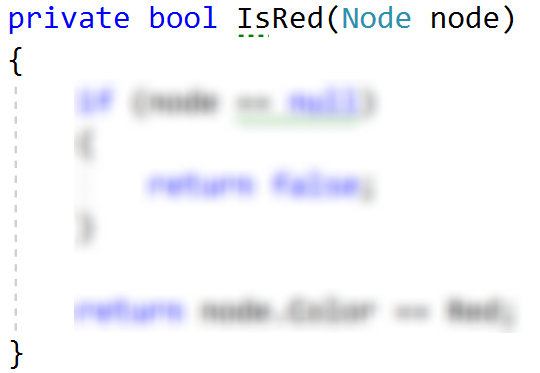


## Check Red Node

Add the following constants to your **RedBlackTree** class:

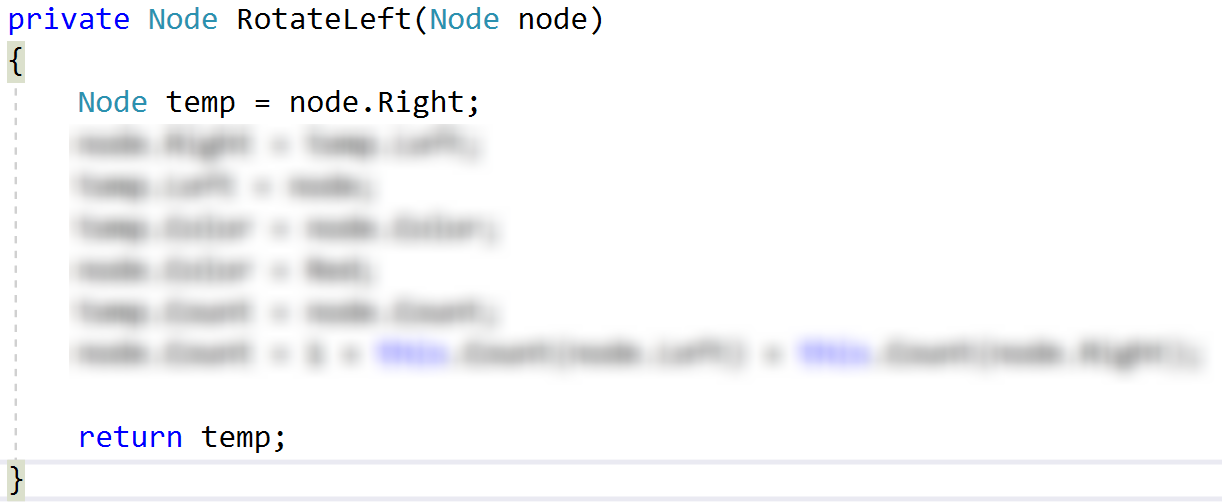


Now create a helper method that will check if a node is red:



## Left Rotation

Create a method that will accomplish the left rotation for a given node.



## Right Rotation

Create a method that will perform right rotation on a given node. The code is similar to the left rotation.

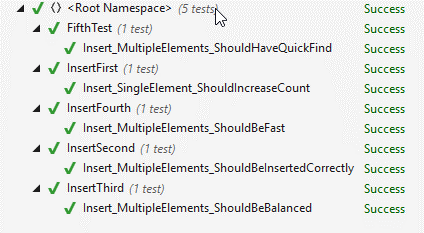
## Flip Colours

Implement a method that will make a node "**red**" and its children "**black**".

## Insert

Modify the existing **Insert()** method. It should create a new **red** node for every insert, **balance** the tree and **recolour** the nodes if needed.

## Run Unit Tests



That's it, you're ready to start implementing delete. :)