

Review of Dmitry Pilyuk by Danil Yevdokimov

Block 2

version 20.04.2022

1 Дерево

1.1 Структура проекта

Норматив структуры проекта на git соблюден. Проект разделен на разные файлы.

1.2 Структура дерева

Есть дополнительное поле Root, а поля First и Last вообще не используются. Не знаю можно ли так.

Нет всевозможны Add..., есть только Add. С Remove, Find та же ситуация. Clear неполный, First и Last не очищаются.

1.3 Структура расширений

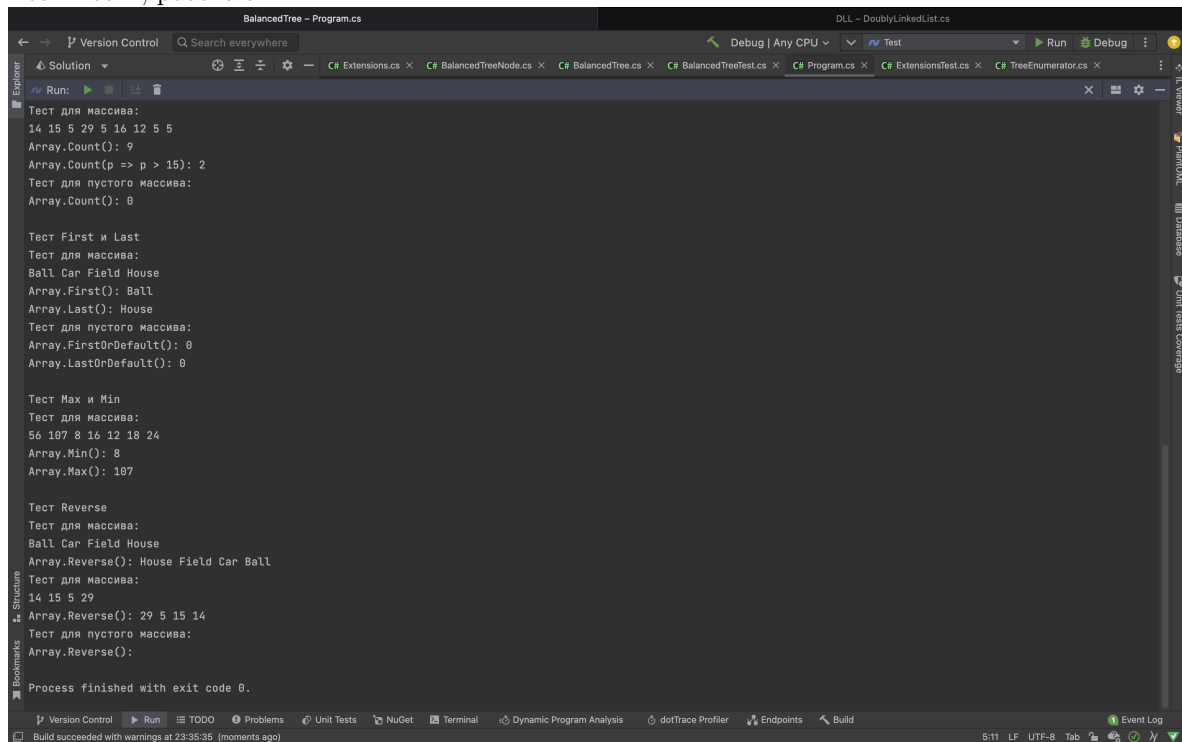
Все обязательные методы реализованы.

1.4 Структура тестов

Не выполнено требование: "Написать тестирующую программу вызывающую эти тестовые функции по параметрам считанным из тестового файла. К проекту приложить не менее трех тестовых файлов". Придется писать свою тестовую функцию для рецензии.

1.5 Тесты

Тесты есть, работают.



The screenshot shows the Visual Studio IDE with the 'Test' window open. The test results are as follows:

```
Test для массива:
14 15 5 29 5 16 12 5 5
Array.Count(): 9
Array.Count(p => p > 15): 2
Test для пустого массива:
Array.Count(): 0

Test First и Last
Test для массива:
Ball Car Field House
Array.First(): Ball
Array.Last(): House
Test для пустого массива:
Array.FirstOrDefault(): 0
Array.LastOrDefault(): 0

Test Max и Min
Test для массива:
56 107 8 16 12 18 24
Array.Min(): 8
Array.Max(): 107

Test Reverse
Test для массива:
Ball Car Field House
Array.Reverse(): House Field Car Ball
Test для массива:
14 15 5 29
Array.Reverse(): 29 5 15 14
Test для пустого массива:
Array.Reverse():

Process finished with exit code 0.
```

Мой тест для деревьев.

```
Yevdokimov Test for Tree
False True True
False True True
Деревья одинаковые? : True
True; True
False; True
Теперь деревья одинаковые? : False
Содержит ли первое дерево false? : False
Содержит ли первое дерево true? : True
```

```
public static void Yevdokimov_Test()
{
    Console.WriteLine("Yevdokimov Test for Tree");
    var tree1 = new BalancedTree<bool>();
    var tree2 = new BalancedTree<bool>(elements: new bool[] {true, false, true});
    tree1.Add(obj: true);
    tree1.Add(obj: new BalancedTreeNode<bool>(data: false));
    tree1.Add(obj: true);
    foreach (var item:bool in tree1) { Console.WriteLine($"{item} "); }
    Console.WriteLine();
    foreach (var item:bool in tree2) { Console.WriteLine($"{item} "); }
    Console.WriteLine();
    Console.WriteLine($"Деревья одинаковые? : {tree1.Equals(tree2)}");
    tree1.Remove(obj: false);
    tree2.Remove(obj: new BalancedTreeNode<bool>(data: true));
    Console.WriteLine(tree1);
    Console.WriteLine(tree2);
    Console.WriteLine($"Теперь деревья одинаковые? : {tree1.Equals(tree2)}");
    Console.WriteLine($"Содержит ли первое дерево false? : {tree1.Contains(data: false)}");
    Console.WriteLine($"Содержит ли первое дерево true? : {tree1.Contains(data: true)}");
    tree1.Clear();
    tree2.Clear();
}
```

Мой тест для расширений.

```
4 1 5 9 Reverse:
9 5 1 4
Array.Min(): 1
Array.Max(): 9
Array.First(): 4
Array.Last(): 9
Array.FirstOrDefault(): 4
Array.LastOrDefault(): 9
Array.All(p => p % 2 == 1): False
Array.Count(): 4
Array.Count(p => p > 4): 2
```

```
public static void YevdokimovTest()
{
    long[] testLongArr = new long[] { 4, 1, 5, 9 };
    foreach (var item :long in testLongArr) { Console.Write($"{item} "); }
    Console.WriteLine("Reverse: ");
    foreach (var item :long in testLongArr.Reverse()) { Console.Write($"{item} "); }
    Console.WriteLine();
    Console.WriteLine($"Array.Min(): {testLongArr.Min()}");
    Console.WriteLine($"Array.Max(): {testLongArr.Max()}");
    Console.WriteLine($"Array.First(): {testLongArr.First()}");
    Console.WriteLine($"Array.Last(): {testLongArr.Last()}");
    Console.WriteLine($"Array.FirstOrDefault(): {testLongArr.FirstOrDefault()}");
    Console.WriteLine($"Array.LastOrDefault(): {testLongArr.LastOrDefault()}");
    Console.WriteLine($"Array.All(p => p % 2 == 1): {testLongArr.All(p :long => p % 2 == 0)}");
    Console.WriteLine($"Array.Count(): {testLongArr.Count()}");
    Console.WriteLine($"Array.Count(p => p > 4): {testLongArr.Count(p :long => p > 4)}");
}
```

2 Добиться ошибок 9 разных типов и их выловить

Его нет.

3 Максимально обобщенная сортировка

3.1 Структура класса сортировок

Есть разные виды сортировок.

3.2 Структура тестов

Мало вариантов использования функций. Надо 9, а есть 5 (без учета, что для каждого алгоритмы сортировки следовало бы иметь свои 9 вариантов).

3.3 Тесты

Мои тесты и встроенные тесты. Мои тесты отмечены пометкой "Yevd."

```
Сортировка пузырьком для массива int
Результат сортировки для массива{36, 198, 24, 16, 11}: {16, 24, 36, 198, 11}
Результат сортировки для массива{}: {}
Результат сортировки для массива{13, 14, 21, 26, 71}: {13, 14, 21, 26, 71}
Результат сортировки для массива{13, 5}: {13, 5}
Yevd. Результат сортировки для массива{8, 2, 3, 10, 1, 22, 1}: {1, 2, 3, 8, 10, 22, 1}
QuickSort для массива char
Результат сортировки для массива{c, d, y, s, e, s, y, u, i, h, f, s, e, f, 2, 4, c}: {2, 4, c, c, d, e, e, f, f, h, i, s, s, u, y, y}
Результат сортировки для массива{a, b, c, d, e}: {a, b, c, d, e}
Результат сортировки для массива{}: {}
Результат сортировки для массива{y, h, a, d}: {a, d, h, y}
Yevd. Результат сортировки для массива{g, o, r, i, k, h, o, v, s, k, i, i}: {g, h, i, i, i, k, k, o, o, r, s, v}
QuickSort для массива string в указанном диапазоне
Результат сортировки для массива{Metallica, One, Blackened, Fuel, Hardwired, Orion, Helpless, Astronomy}: {Metallica, Astronomy, Blackened, Fuel, Hardwired, Helpless, One, Orion}
Результат сортировки для массива{}: {}
Результат сортировки для массива{Dog, Forest, House, Car, Man, Dream}: {Car, Dog, Forest, House, Man, Dream}
Результат сортировки для массива{Abba, Beatles, queen, Queen, Soundgarden}: {Abba, Beatles, queen, Queen, Soundgarden}
Yevd. Результат сортировки для массива{Abudabi, Arab, Hinkali, Hi}: {Abudabi, Arab, Hi, Hinkali}
Сортировка пузырьком для массива int с использованием компаратора
Результат сортировки для массива{25, 162, 12, 9384, 2, 54, 72, 18, 53, 72}: {9384, 162, 72, 72, 54, 53, 25, 18, 12, 2}
Результат сортировки для массива{22, 12, 9, 3}: {22, 12, 9, 3}
Результат сортировки для массива{}: {}
Результат сортировки для массива{1, 12}: {12, 1}
Yevd. Результат сортировки для массива{8, 2, 3, 10, 1, 22, 1}: {22, 10, 8, 3, 2, 1, 1}
QuickSort для массива bool
Результат сортировки для массива{True, False, False, True}: {False, False, True, True}
Результат сортировки для массива{}: {}
Результат сортировки для массива{True, True, False}: {False, True, True}
Результат сортировки для массива{False, False, True, True}: {False, False, False, True, True}
Yevd. Результат сортировки для массива{True, True, False, False, True}: {False, False, True, True, True}
QuickSort для массива Person
Результат сортировки для массива{Alex: 1000, Tom: 300, John: 550, May: 700}: {Tom: 300, John: 550, May: 700, Alex: 1000}
Результат сортировки для массива{}: {}
Результат сортировки для массива{Alex: 1000, Tom: 2000, John: 3000, May: 4000}: {Alex: 1000, Tom: 2000, John: 3000, May: 4000}
Результат сортировки для массива{Alex: 700, Tom: 600, John: 200, May: 50}: {May: 50, John: 200, Tom: 600, Alex: 700}
Yevd. Результат сортировки для массива{Alex: 100, Tom: 800, John: 10, May: 50}: {John: 10, May: 50, Alex: 100, Tom: 800}
```