

(3 балла)

Создайте параметризованный класс двоичного дерева поиска, позволяющий хранить объекты, реализующие интерфейс `java.lang.Comparable`. Дерево должно быть реализовано таким образом, что в левой ветке хранятся узлы со значением меньше корня, а в правой - больше корня. В классе должны быть реализованы следующие методы:

1. добавление узла (при добавлении элемента, значение которого уже присутствует в дереве должно быть сгенерировано Ваше собственное исключение `BinaryTreeInsertionException`)
2. нахождение высоты дерева (возвращает `int`)
3. поиск элемента (возвращает `boolean`)

(3 балла)

Создайте параметризованный интерфейс `Selector<T>`, содержащий следующие методы:

1. `T current()` - возвращает текущий элемент. Можно вызывать сколько угодно раз.
2. `boolean hasNext()` - есть ли следующий
3. `void next()` - перейти к следующему

Добавьте к классу дерева два метода, один возвращает `Selector`, перебирающий элементы от меньшего к большему, а другой - от большего к меньшему, соответственно.

(3 балла)

Сделайте так, чтобы эти `Selector`'ы корректно работали в том случае, если после создания `Selector`'а и прохождения нескольких элементов в дерево были добавлены узлы. Такие узлы должны быть выданы селектором в "естественном" порядке, т.е., например, селектор перебирающий элементы от меньшего к большему, выдает минимальный элемент из не обработанных.

Например, к дереву с корнем 5 добавили узлы 2, 0, 4, 8, 6, 10. После чего, был создан селектор и с его помощью пройдены узлы 0, 2, 4, 5, 6. Если после этого в дерево добавили узлы 1, 3, 7, то селектор следующими выдаст 1, 3, 7, 8, 10.

`Selector`'ы должны быть реализованы таким образом, чтобы время работы операций было как можно меньшим. В идеале следующим:

1. `current` - $O(1)$
2. `hasNext`, `next` - в лучшем случае $O(1)$, в худшем $O(\text{высоты дерева})$

(1 балл)

Создайте непараметризованный класс, содержащий статический параметризованный метод, принимающий на вход Selector и печатающий все элементы контейнера, к которому относится переданный Selector.

(1 балл)

Ваша программа должна принимать в качестве параметра имя файла, содержащего целые числа. Эти числа должны добавляться в дерево в порядке их следования в файле. Программа должна выводить на экран эти числа от меньшего к большему и от большего к меньшему, используя описанные выше методы.