

## Klasa Świra

Uwaga:

- do zaimplementowania klasa o nazwie `a` w pakiecie domyślnym i pliku `"a.java"`
- niedozwolone jest definiowanie klas o nazwach zaczynających się słowem `Test`
- niedozwolone jest używanie pól statycznych
- **poniższe zadanie prezentuje wyjątkowo złe techniki programistyczne i służy wyłącznie do sprawdzenia znajomości pewnych aspektów Javy**

Pewien informatyk, powszechnie uważany za świra, postanowił spróbować swoich sił w pisaniu w Javie. Jego pierwszym (i ostatnim) zadaniem w nowej pracy było zaimplementowanie klasy, o wdzięcznej nazwie `"a"`, służącej do przechowywania liczb całkowitych (`int`'ów).

Klasa miała:

- przechowywać listę `int`'ów
- umożliwiać wstawianie zarówno na początek, jak i na koniec listy
- wyświetlać elementy w kolejności w jakiej są w liście i w kolejności naturalnego porządku
- umożliwiać `pop()` (od tyłu listy)

(Jako, że świr nie miał opinii dobrego programisty, pracodawca sprecyzował zadanie żądając aby tylko wstawianie elementów i wspomniany wyżej `pop` modyfikowały listę.)

Oryginalna implementacja klasy świra zaginęła, a Twoim zadaniem jest ją odtworzyć na podstawie testów unitarnych które po niej pozostały (test11 wymaga zajrzenia do dokumentacji Javy - interfejs `Iterable`):

Uwaga: plik z rozwiązaniem powinien mieć nazwę **`a.java`** !

```
import static org.junit.Assert.*;
import java.util.Arrays;
import org.junit.Test;

public class aTest {

    @Test
    public void test0() {
        a a = new a();
        a.a(1);
        a.a(2);
        a.a(3);
        assertEquals(String.valueOf(a), "[1, 2, 3]");
    }

    @Test
    public void test1() {
        a a = new a();
        a.a(1);
```

```
        a.a(3);
        a.a(2);
        assertEquals(String.valueOf(a), "[1, 2, 3]");
    }

    @Test
    public void test2() {
        a a = new a();
        a.a(1).a(3).a(2);
        assertEquals(String.valueOf(a), "[1, 2, 3]");
        assertEquals(String.valueOf(a.a), "[1, 3, 2]");
    }

    @Test
    public void test3() {
        a a = new a();
        a.a(1).a(5).a(2).a(4).a(3);
        assertEquals(String.valueOf(a), "[1, 2, 3, 4, 5]");
        assertEquals(String.valueOf(a.a), "[1, 5, 2, 4, 3]");
        assertEquals(String.valueOf(a.a()), "3");
        assertEquals(String.valueOf(a), "[1, 2, 4, 5]");
        assertEquals(String.valueOf(a.a), "[1, 5, 2, 4]");
        assertEquals(String.valueOf(a.a()), "4");
        assertEquals(String.valueOf(a), "[1, 2, 5]");
        assertEquals(String.valueOf(a.a), "[1, 5, 2]");
        assertEquals(String.valueOf(a.a()), "2");
        assertEquals(String.valueOf(a), "[1, 5]");
        assertEquals(String.valueOf(a.a), "[1, 5]");
        assertEquals(String.valueOf(a.a()), "5");
        assertEquals(String.valueOf(a), "[1]");
        assertEquals(String.valueOf(a.a), "[1]");
        assertEquals(String.valueOf(a.a()), "1");
        assertEquals(String.valueOf(a), "[]");
        assertEquals(String.valueOf(a.a), "[]");
    }

    @Test
    public void test4() {
        a a = new a();
        a.a(1).a(5).a(2).a(4).a(3);
        assertEquals(String.valueOf(a), "[1, 2, 3, 4, 5]");
        assertEquals(String.valueOf(a.a), "[1, 5, 2, 4, 3]");
        a.a(a.a());
        assertEquals(String.valueOf(a), "[1, 2, 3, 4, 5]");
        assertEquals(String.valueOf(a.a), "[3, 1, 5, 2, 4]");
        a.a(a.a());
        assertEquals(String.valueOf(a), "[1, 2, 3, 4, 5]");
        assertEquals(String.valueOf(a.a), "[4, 3, 1, 5, 2]");
        a.a(a.a()).a(a.a()).a(a.a());
        assertEquals(String.valueOf(a), "[1, 2, 3, 4, 5]");
        assertEquals(String.valueOf(a.a), "[1, 5, 2, 4, 3]");
        a.a(1).a(5);
    }
```

Klasa Świra

```
        a = a.a;
    }

    }

    @Test
    public void test9() {
        a a = new a();
        assertEquals(String.valueOf(a.a(1).a(3).a(2)), "[1, 2, 3]");
        a = new a();
        assertEquals(String.valueOf(a.a(1).a(3).a(2).a), "[1, 3, 2]");
        a = new a();
        assertEquals(String.valueOf(a.a.a(1).a(3).a(2)), "[1, 2, 3]");
        a = new a();
        assertEquals(String.valueOf(a.a.a(1).a.a.a(3).a.a.a.a.a(2)), "[1, 2, 3]");
    }

    @Test
    public void test10() {
        a a = new a();
        assertEquals(String.valueOf(a.a(1).a(2).a(3).a(4).a(5)), "[1, 2, 3, 4, 5]");
        assertEquals(String.valueOf(a.a(a.a())),"[5, 1, 2, 3, 4]");
        assertEquals(String.valueOf(a.a(a.a())),"[4, 5, 1, 2, 3]");
        assertEquals(String.valueOf(a.a(6)), "[1, 2, 3, 4, 5, 6]");
        assertEquals(String.valueOf(a.a(a.a())),"[6, 4, 5, 1, 2, 3]");
    }

    @Test
    public void test11() {
        String[] one = {"1", "3", "2"};
        String[] two = {"1", "2", "3"};
        a o = new a();
        int i = 0;
        o.a(1);
        o.a(3);
        o.a(2);
        for (a a : o)
            assertEquals(String.valueOf(a), one[i++]);
        i = 0;
        for (a a : o.a)
            assertEquals(String.valueOf(a), two[i++]);
    }
}
```