

## Task 2

1. Задано довільний рядок виду `"isu([syvstc]ts(crs))cs"`, в якому довільним чином можуть бути розставлені круглі або квадратні дужки, що відкриваються або закриваються. Напишіть функцію `testString`, яка визначатиме коректність розміщення дужок (повертає `true` або `false`). Під коректністю мається на увазі відсутність «перетину» різних видів дужок (наприклад, так `isu[syv(stc]ts(crs))cs"`) або відповідність кожній дужці, що відкривається такого ж типу, але такої, що закривається.

2. Написати метод `substrCount`, який повертає кількість входжень підрядка. Метод приймає такі аргументи: `input` – рядок, у якому ведеться пошук, `needle` – підрядок, що шукається, `offset` - зміщення початку відліку, `length` – максимальна довжина рядка, у якій буде здійснюватися пошук підрядка після зазначеного зсуву. Приклад виклику: `substrCount('Good Golly Miss Molly', 'll', 7, 10)`.

3. Написати метод, що реалізує гру "Вгадай число". Комп'ютер загадує ціле число у діапазоні від 1 до 100 и просить користувача вгадати. Користувач вводить число, якщо воно більше ніж загадане, виводиться повідомлення "Багато", якщо менше - "Менше" і знову очікується введення числа. Якщо користувач вгадав, гра закінчується та виводиться повідомлення "Молодець!". Число комп'ютер загадує один раз, після кожної спроби його не треба перевизначати. Число генерувати за допомогою класу `Random` з пакету `java.util`.

4. Створити та розташувати в кореневому пакеті (`ua.com.sourceit.your_last_name.subtask4`) інтерфейс `Container`:

-----

```

package ua.com.sourceit.your_last_name.subtask4;

public interface Container extends Iterable<Object> {

    // Removes all of the elements.
    void clear();

    // Returns the number of elements.
    int size();

    // Returns a string representation of this container.
    String toString();

    // Returns an iterator over elements.
    // Iterator must implements the remove method.
    Iterator<Object> iterator();
}

```

-----

Зауваження. Заборонено використовувати будь-які типи з пакету java.util, крім двох:

- 1) java.util.Iterator
- 2) java.util.NoSuchElementException

-----

Имена типов: Array, ArrayImpl

-----

4.1. Створити інтерфейс Array наступного змісту:

-----

```

package ua.com.sourceit.your_last_name.subtask4;

public interface Array extends Container {

    // Add the specified element to the end.
    void add(Object element);

    // Sets the element at the specified position.
    void set(int index, Object element);

    // Returns the element at the specified position.

```

```

    Object get(int index);

    // Returns the index of the first occurrence of the specified element,
    // or -1 if this array does not contain the element.
    // (use 'equals' method to check an occurrence)
    int indexOf(Object element);

    // Removes the element at the specified position.
    void remove(int index);

}

```

-----

4.2. Створити клас `ArrayImpl`, що реалізує інтерфейс `Array`.

Зберігання об'єктів усередині контейнера реалізувати за допомогою масиву об'єктів.

Метод `iterator` повинен повертати екземпляр класу `IteratorImpl`, який реалізує інтерфейс `java.util.Iterator<Object>`.

Клас `IteratorImpl` має бути визначений усередині класу `ArrayImpl` (є внутрішнім класом).

Якщо у контейнер були додані за допомогою методу `add` три елементи A, B, C, то:

- 1) метод `toString` повинен повертати рядок "[A, B, C]"
- 2) порядок обходу елементів контейнера ітератором: A B C

4.3. У класі `ArrayImpl` створити метод `main`, в якому продемонструвати роботу:

- 1) всіх методів інтерфейсу `Array` (включаючи успадковані від `Container` та `Iterable`);
- 2) всіх методів інтерфейсу `Iterator` (`hasNext`/`next`/`remove`).