لیست پیوندی

لیست یکی از ساختمان داده هایی است که برای نگهداری عناصر استفاده میشود. یکی از انواع لیست ، لیست پیوندی یا همان LinkedList معروف است که احتمالا کمابیش با آن آشنا هستید و در ادامه درس با آن بیشتر آشنا خواهید شد.

در این تمرین از شما میخواهیم که چیزی شبیه LinkedList که عدد صحیح نگه می‌دارد را پیاده سازی کنید. داکیومنت‌های کلاس LinkedList را می‌توانید اینجا بخوانید.

شما باید کلاس LinkedList را **خودتان** و با توجه به توضیحات زیر پیاده سازی کنید.

برنامه شما باید شامل همه متد های زیر باشد.

```
/***
* adds the specified element to the end of the list
***/
public boolean add(Integer element) ;

/***
* adds the specified element
* at the specified index of your list
***/
public void add(int index , Integer element);

/***
* append all of the elements
* in the specified linkedlist to the end of this list
***/
public boolean addAll(LinkedList linkedlist);


/**appends all of the elements
* in the specified linkedlist starting at the specified index
***/
public boolean addAll(int index, LinkedList linkedlist);

```

```
24  /***
25  * inserts a specified element at the beginning of this list
26  ***/
27  public void addFirst(Integer element);
28
29  /***
30  * appends the specified element at the end of this list
31  ***/
32  public void addLast(Integer element);
33
34  /***
35  * Removes all of the elements from this list. The list will be empty after this
36  ***/
37  public void clear();
38
39  /***
40  * Returns `true` if this list contains the specified element.
41  ***/
42  public boolean contains(Integer i);
43
44  /***
45  * Returns the element at the specified position in this list.
46  ***/
47  public Integer get(int index);
48
49  /***
50  * Returns the index of the first occurrence of the specified element in this lis
51  ***/
52  public int indexOf(Integer i);
53
54  /***
55  * Retrieves and removes the head (first element) of this list.
56  ***/
57  public Integer remove();
58
59  /***
60  * Removes the element at the specified position in this list.
61  * Shifts any subsequent elements to the left (subtracts one from their indices).
62  *  Returns the element that was removed from the list.
63  ***/
```

```
public Integer remove(int index);

/***
* Returns the number of elements in this list.
***/
public int size();

/***
* Returns an array containing all of the elements in this list in proper sequenc
***/
public Integer[] toArray();

/***
* Returns true if this list contains no elements.
***/
public boolean isEmpty();
```

کد نمونه

```
public class Main {
    public static void main(String[] args) {
        LinkedList linkedList1 = new LinkedList();
        linkedList1.add(1);
        linkedList1.add(3);
        System.out.println(linkedList1.size()); // 2
        linkedList1.add(1, 2);
        Integer list[] = linkedList1.toArray();
        for (int i = 0; i < list.length; i++) {
            System.out.println(list[i]); // 1 2 3
        }
        System.out.println(linkedList1.contains(4)); // false
        System.out.println(linkedList1.indexOf(1)); // 0
        LinkedList linkedList2 = new LinkedList();
        linkedList2.addAll(linkedList1);
        System.out.println(linkedList2.size()); // 3
        linkedList2.clear();
        System.out.println(linkedList2.isEmpty()); // true

```

```
20      }
    }
```

توجه مهم

این سوال به صورت اتوماتیک و متفاوت با سوالات دیگری که تا کنون دیده‌اید داوری می‌شود. برای داوری از JUnit استفاده شده است. برای داوری صحیح، موارد گفته شده مثل نام متدها و کانستراکتور و غیره را دقیقا در کلاس قرار دهید.

نهایتا یک فایل زیپ اپلود کنید که فقط شامل LinkedList.java باشد.

در صورتی که به عبارت could not run 9 tests. برخوردید به معنی کامپایل ارور و رعایت نکردن موارد فوق است.

محتویات فایل زیر مثل تصویر زیر باشد:

```
.
└── LinkedList.java

0 directories, 1 file
```

اندرومدا

آیا به نجوم علاقه دارید؟ تا کنون به آسمانِ شب نگاه کرده‌اید؟ تعداد ستاره‌های کهکشان چقدر است؟ تعداد مولکول‌های کهکشان اندرومدا چقدر است؟ آیا این تعداد در integer جا می‌شود؟ در long چطور؟ احتمالا جواب منفی است. برای نگه‌داشتن اعداد بسیار بزرگ، برنامه‌نویسان جاوا از کلاس‌های BigDecimal و یا BigInteger استفاده می‌کنند.

داکیومنت‌های کلاس BigInteger را می‌توانید اینجا بخوانید.

حالا ناسا برای نیازهای خاص و محرمانه خودش، از شما خواسته که کلاس BigInteger را با خواص زیر **خودتان از ابتدا با استفاده از لیست پیوندی** پیاده کنید. استفاده از ۲ کلاس BigDecimal و BigInteger آمادهٔ جاوا و هر کد آماده‌ای برای کار با اعداد بزرگ نمره‌ای ندارد.

چیزهایی که باید رعایت کنید:

- نام کلاس شما باید BigInteger باشد.
- سازندهٔ گفته‌شده
- متدهای گفته شده
- فیلدهای استاتیکی که ذکر می شود
- کلاس باید **immutable** باشد، یعنی بعد از construct شدن، فیلدهایش هیچ تغییری نکند.
- برای پیاده‌سازی فقط از لیست پیوندی استفاده کنید. (یعنی نه آرایه و نه String)
- می‌توانید از کلاس لیست پیوندی جاوا (java.util.LinkedList) استفاده کنید ولی استفاده از لیست پیوندی خودتان نمره امتیازی دارد.

سازندهٔ زیر را عینا پیاده‌کنید.

```
1 /**
2 * Translates the decimal String representation of a
3 * BigInteger into a BigInteger.
4 */
5 public BigInteger(String val)
```

متدهای زیر را دقیقا پیاده‌سازی کنید.

```java
/**
* Returns a BigInteger whose value is the absolute
* value of this  BigInteger.
*/
public BigInteger abs()

/**
* Returns a BigInteger whose value is
* (this + val).
*/
public BigInteger add(BigInteger val)

/**
* Returns a BigInteger whose value is (this - val).
*/
public BigInteger subtract(BigInteger value)

/**
* Returns a BigInteger whose value is (this * val).
*/
public BigInteger multiply(BigInteger value)

/**
* Compares this BigInteger with o for equality.
* if o is not BigInteger, return false
*/
public boolean equals(Object o)

/**
* Returns the decimal String representation
* of this BigInteger.
*/
public String toString()
```

فیلدهای استاتیک زیر باید در کلاس شما وجود داشته باشند و مقداردهی شده باشند.

```
public static final BigInteger ONE;
public static final BigInteger TEN;
public static final BigInteger ZERO;
```

توجه

- تمامی توابع از جمله جمع، نباید چیزی درون شی را عوض کنند، مثلا a.add(b) باعث می‌شود مقدار a+b برگردانده شود و a و b خودشان هیچ تغییری نمی‌کنند.
- همان‌طور که می‌دانید یکی از مزایای لیست‌پیوندی این است که تمام داده‌ها را پشت‌سر هم داخل رم نگه نمی‌دارد و این برای زمانی که داده‌های زیادی داریم مثل این سوال، مزیت مهمی محسوب می‌شود.
- از لیست‌پیوندی جاوا می‌توانید مثل کد زیر استفاده کنید:

```

/**
* example of using linked list of Integer
* you should import java.util.LinkedList for this
*/
LinkedList<Integer> myList = new LinkedList<Integer>();
 myList.add(1);
 myList.add(2);
 myList.add(3);
 myList.addFirst(0);
 int len = myList.size();
System.out.println(myList.get(len-1)); // print last item:0
```

کد نمونه

```
    public static void main(String[] args){

        /**
            example of construct by String
        */
        BigInteger b1 = new BigInteger("+1234");
```

```
7           System.out.println(b1.toString()); //1234
8
9           BigInteger b2 = new BigInteger("3333");
10          System.out.println(b2.toString()); //3333
11
12          BigInteger b3 = new BigInteger("-10");
13          System.out.println(b3.toString()); //-10
14
15
16          /**
17          * example of add
18          */
19          BigInteger bz = b3.add(BigInteger.TEN);
20          System.out.println(bz.toString()); // -10 + 10 = 0
21
22          /**
23          * example of other operations
24          */
25          System.out.println( b2.add(b3.abs()).toString() ); //3343
26          //b3.abs() will compute to 10
27          // b2+10 will be 3343
28
29          System.out.println(new BigInteger("7").multiply(new BigInteger("-4")).to
30
31          System.out.println(new BigInteger("74").subtract(new BigInteger("-34")).
32
33
34          System.out.println(b1.equals(b2)); // b1!=b2 ==> false
35
36          System.out.println(bz.equals(BigInteger.ZERO)); // both zero ==> true
37
38          System.out.println(bz.equals(0)); // return false, because 0 is Integer
39      }
```

کد نمونه و امضای توابع را می‌توانید از این لینک دانلود کنید.

توجه مهم: برای اینکه کد شما کامپایل شود شما باید امضای توابع را حفظ کنید، در صورتی که کوچک‌ترین تغییری بدهید، کد شما کامپایل نشده و نمره‌ای دریافت نمی‌کنید.

همچنین پیشنهاد اکید می‌شود همین فایل BigInteger را دانلود کرده و پیاده‌سازی‌ها را داخلش انجام دهید.

ضمنا پیاده‌سازی تابع equals و toString را هم در اولویت قرار دهید چون تست این سوال بر اساس این ۲ تابع کار می‌کند.

این سوال به صورت اتوماتیک و متفاوت با سوالات دیگری که تا کنون دیده‌اید داوری می‌شود. برای داوری از JUnit استفاده شده است. برای داوری صحیح، موارد گفته شده مثل نام متدها و کانستراکتور و غیره را دقیقا در کلاس قرار دهید.

نهایتا یک فایل زیپ اپلود کنید که فقط شامل BigInteger.java باشد. در صورتی که از لینک لیست خودتان استفاده می‌کنید می‌توانید (اختیاری) یک فایل LinkedList.java هم در فایل زیپ داشته باشید.

در صورتی که به عبارت could not run 12 tests. برخوردید به معنی کامپایل ارور و رعایت نکردن موارد فوق است.

محتویات فایل زیپ به یکی از ۲ حالت زیر باید باشد:

```
.
├── BigInteger.java
└── LinkedList.java

0 directories, 2 files
```

```
.
└── BigInteger.java

0 directories, 1 file
```