IIP

Test Unit 7 - Possible solution

Year 2014-2015

Name:

1. The golden ratio number $\phi = \frac{1+\sqrt{5}}{2}$ could be approximated by the quotient of two successive terms of the Fibbonacci's succession (i.e., $\frac{t_n}{t_{n-1}}, n > 1$). Based on Fibbonacci's succession (t$_1$=1, t$_2$=1, t$_i$=t$_{i-2}$+t$_{i-1}$, $i > 2$), write a Java class method that receives a tolerance error `epsilon` (positive real value lower than 1) and returns an approximation to ϕ with an error lower than `epsilon`. Error in the approximation is always lower than the absolute value of the difference on the approximations obtained by term n and by term $n-1$ (i.e., $\left|\frac{t_{n-1}}{t_{n-2}} - \frac{t_n}{t_{n-1}}\right|, n > 2$).

```
public static double goldenNumber(double epsilon) {
  double p=1, c=1, n, err=1;

  while (err>=epsilon) {
    n=p+c;
    err=Math.abs(c/p-n/c);
    p=c;
    c=n;
  }

  return c/p;
}
```

2. Implement a Java class method that receives a `String` and returns the reverse `String` (e.g., for "Java" will return "avaJ").

```
public static String reverse(String s) {
  String res="";
  int i=0;

  for (i=0;i<s.length();i++) res=s.charAt(i)+res;

  return res;
}
```

3. Write a Java program class whose `main` method draws the following figure:

```
   *
  * *
 *   *
*     *
```

The number of asterisks of a diagonal must be asked for the user (in the previous figure is 4). You can suppose that the user will always input a value greater or equal than 1.

```
import java.util.*;

public class Roof {
  public static void main(String [] args) {
    Scanner kbd=new Scanner(System.in).useLocale(Locale.US);
    int i, j, n;

    System.out.print("n = "); n=kbd.nextInt();

    for (j=0;j<(n-1);j++) System.out.print(" ");
    System.out.println("*");

    for (i=1;i<n;i++) {
      for (j=0;j<(n-i-1);j++) System.out.print(" ");
      System.out.print("*");
      for (j=0;j<(i*2-1);j++) System.out.print(" ");
      System.out.println("*");
    }
  }
}
```