```java
/**
 * Méthodes standard sur des nombres entiers.
 *
 * @author      biech153 (Biersbach Chris) / gamca174 (Gamboa Carlos) / olial319 (Olinger Alex)
 * @version     23/05/2019 7:00:50
 * Classe:      3GIG
 */
public class IntegerNumber
{
     private int number;

     public IntegerNumber(int pNumber)
     {
          number = Math.abs(pNumber);
     }

     public int getNumber()
     {
          return number;
     }

     public boolean isEven()
     {
          // version 1
          if (number%2 == 0)
               return true;
          else
               return false;

          // version 2 (mieux)
          // return number%2==0;
     }

     public boolean isPrime()
     {
          int count = 0;
          for (int i=1; i<=number; i++)
          {
               if (number%i == 0)
                    count++;
          }

          if (count == 2)
               return true;
          else
               return false;

          // ou encore mieux:
          //   return (count == 2);
     }

     public int sumOfDividers(int pNumber)
     {
          int sumDiv=0;
          for (int i=1; i<=pNumber; i++)
          {
               if (pNumber%i == 0)
                    sumDiv = sumDiv + i;
          }
          return sumDiv;
     }

     public int sumOfDividers()
     {
          return sumOfDividers(number);
     }

     public boolean isFriendlyTo(int pOtherNumber)
     {
          int sum = sumOfDividers();
          int sumOtherNumber = sumOfDividers(pOtherNumber);

          if (sum == sumOtherNumber)
          {
               return true;
          }
          else
          {
               return false;
          }
     }
```

```
    public boolean isPerfect()
    {
        return sumOfDividers() == 2*number;
    }

    public boolean isDeficient()
    {
        return sumOfDividers() < 2*number;
    }

    public boolean isAbundant()
    {
        return sumOfDividers() > 2*number;
    }

    public int reverse()
    {
        int res = 0;
        int val = number;
        while (val!=0)
        {
            res = res*10 + val%10;
            val = val/10;
        }
        return res;
    }

    public boolean isPalindrome()
    {
        int rev = reverse();
        return number==rev;
    }
}
```

```java
/**
 * Teste les méthodes / algorithmes programmés dans IntegerNumber.
 *
 * @author     biech153 (Biersbach Chris) / gamca174 (Gamboa Carlos) / olial319 (Olinger Alex)
 * @version    23/05/2019 7:00:50
 * Classe:     3GIG
 */
public class TestIntegerNumber
{
     /**
      * Programme principal.
      */
     public static void main(String[] args)
     {
          IntegerNumber a;

          System.out.println("*** nombres pairs ou impairs ***");
          a = new IntegerNumber(1); System.out.println("  n="+a.getNumber()+"   --> isEven() = "+a.isEven());
          a = new IntegerNumber(2); System.out.println("  n="+a.getNumber()+"   --> isEven() = "+a.isEven());
          a = new IntegerNumber(3); System.out.println("  n="+a.getNumber()+"   --> isEven() = "+a.isEven());
          a = new IntegerNumber(4); System.out.println("  n="+a.getNumber()+"   --> isEven() = "+a.isEven());
          a = new IntegerNumber(5); System.out.println("  n="+a.getNumber()+"   --> isEven() = "+a.isEven());
          a = new IntegerNumber(6); System.out.println("  n="+a.getNumber()+"   --> isEven() = "+a.isEven());
          a = new IntegerNumber(15); System.out.println("  n="+a.getNumber()+"  --> isEven() = "+a.isEven());
          a = new IntegerNumber(16); System.out.println("  n="+a.getNumber()+"  --> isEven() = "+a.isEven());
          a = new IntegerNumber(17); System.out.println("  n="+a.getNumber()+"  --> isEven() = "+a.isEven());
          a = new IntegerNumber(121); System.out.println("  n="+a.getNumber()+" --> isEven() = "+a.isEven());
          a = new IntegerNumber(122); System.out.println("  n="+a.getNumber()+" --> isEven() = "+a.isEven());
          a = new IntegerNumber(123); System.out.println("  n="+a.getNumber()+" --> isEven() = "+a.isEven());

          System.out.println();
          System.out.println("*** nombres premiers ***");
          a = new IntegerNumber(1); System.out.println("  n="+a.getNumber()+"    --> isPrime() = "+a.isPrime());
          a = new IntegerNumber(2); System.out.println("  n="+a.getNumber()+"    --> isPrime() = "+a.isPrime());
          a = new IntegerNumber(3); System.out.println("  n="+a.getNumber()+"    --> isPrime() = "+a.isPrime());
          a = new IntegerNumber(4); System.out.println("  n="+a.getNumber()+"    --> isPrime() = "+a.isPrime());
          a = new IntegerNumber(5); System.out.println("  n="+a.getNumber()+"    --> isPrime() = "+a.isPrime());
          a = new IntegerNumber(6); System.out.println("  n="+a.getNumber()+"    --> isPrime() = "+a.isPrime());
          a = new IntegerNumber(15); System.out.println("  n="+a.getNumber()+"   --> isPrime() = "+a.isPrime());
          a = new IntegerNumber(16); System.out.println("  n="+a.getNumber()+"   --> isPrime() = "+a.isPrime());
          a = new IntegerNumber(17); System.out.println("  n="+a.getNumber()+"   --> isPrime() = "+a.isPrime());
          a = new IntegerNumber(121); System.out.println("  n="+a.getNumber()+"  --> isPrime() = "+a.isPrime());
          a = new IntegerNumber(122); System.out.println("  n="+a.getNumber()+"  --> isPrime() = "+a.isPrime());
          a = new IntegerNumber(123); System.out.println("  n="+a.getNumber()+"  --> isPrime() = "+a.isPrime());

          System.out.println();
          System.out.println("*** calculs sur la somme des diviseurs ***");
          a = new IntegerNumber(1); System.out.println("  n="+a.getNumber()+"    / sumOfDividers() = "+a.sumOfDividers()+"     / 2*n
= "+(a.getNumber()*2)+"     / isPerfect() = "+a.isPerfect()+" / isDeficient() = "+a.isDeficient()+"  / isAbundant() = "+a.isAbundant());
          a = new IntegerNumber(2); System.out.println("  n="+a.getNumber()+"    / sumOfDividers() = "+a.sumOfDividers()+"     / 2*n
= "+(a.getNumber()*2)+"     / isPerfect() = "+a.isPerfect()+" / isDeficient() = "+a.isDeficient()+"  / isAbundant() = "+a.isAbundant());
          a = new IntegerNumber(3); System.out.println("  n="+a.getNumber()+"    / sumOfDividers() = "+a.sumOfDividers()+"     / 2*n
= "+(a.getNumber()*2)+"     / isPerfect() = "+a.isPerfect()+" / isDeficient() = "+a.isDeficient()+"  / isAbundant() = "+a.isAbundant());
          a = new IntegerNumber(4); System.out.println("  n="+a.getNumber()+"    / sumOfDividers() = "+a.sumOfDividers()+"     / 2*n
= "+(a.getNumber()*2)+"     / isPerfect() = "+a.isPerfect()+" / isDeficient() = "+a.isDeficient()+"  / isAbundant() = "+a.isAbundant());
          a = new IntegerNumber(5); System.out.println("  n="+a.getNumber()+"    / sumOfDividers() = "+a.sumOfDividers()+"     / 2*n
= "+(a.getNumber()*2)+"    / isPerfect() = "+a.isPerfect()+" / isDeficient() = "+a.isDeficient()+"  / isAbundant() = "+a.isAbundant());
          a = new IntegerNumber(6); System.out.println("  n="+a.getNumber()+"    / sumOfDividers() = "+a.sumOfDividers()+"    / 2*n
= "+(a.getNumber()*2)+"    / isPerfect() = "+a.isPerfect()+"  / isDeficient() = "+a.isDeficient()+" / isAbundant() = "+a.isAbundant());
          a = new IntegerNumber(12); System.out.println("  n="+a.getNumber()+"   / sumOfDividers() = "+a.sumOfDividers()+"    / 2*n
= "+(a.getNumber()*2)+"    / isPerfect() = "+a.isPerfect()+" / isDeficient() = "+a.isDeficient()+" / isAbundant() = "+a.isAbundant());
          a = new IntegerNumber(28); System.out.println("  n="+a.getNumber()+"   / sumOfDividers() = "+a.sumOfDividers()+"    / 2*n
= "+(a.getNumber()*2)+"    / isPerfect() = "+a.isPerfect()+"  / isDeficient() = "+a.isDeficient()+" / isAbundant() = "+a.isAbundant());
          a = new IntegerNumber(496); System.out.println("  n="+a.getNumber()+"  / sumOfDividers() = "+a.sumOfDividers()+"    / 2*n =
"+(a.getNumber()*2)+"   / isPerfect() = "+a.isPerfect()+"  / isDeficient() = "+a.isDeficient()+" / isAbundant() = "+a.isAbundant());
          a = new IntegerNumber(8128); System.out.println("  n="+a.getNumber()+" / sumOfDividers() = "+a.sumOfDividers()+" / 2*n = "+(a.
/ isPerfect() = "+a.isPerfect()+"  / isDeficient() = "+a.isDeficient()+" / isAbundant() = "+a.isAbundant());
          a = new IntegerNumber(8129); System.out.println("  n="+a.getNumber()+" / sumOfDividers() = "+a.sumOfDividers()+"  / 2*n =
"+(a.getNumber()*2)+" / isPerfect() = "+a.isPerfect()+" / isDeficient() = "+a.isDeficient()+"  / isAbundant() = "+a.isAbundant());

          System.out.println();
          System.out.println("*** renversement de valeurs ***");
          a = new IntegerNumber(1); System.out.println("  n="+a.getNumber()+"     / reverse() = "+a.reverse()+"     / isPalindrome()
= "+a.isPalindrome());
          a = new IntegerNumber(2); System.out.println("  n="+a.getNumber()+"     / reverse() = "+a.reverse()+"     / isPalindrome()
= "+a.isPalindrome());
          a = new IntegerNumber(9); System.out.println("  n="+a.getNumber()+"     / reverse() = "+a.reverse()+"     / isPalindrome()
= "+a.isPalindrome());
          a = new IntegerNumber(10); System.out.println("  n="+a.getNumber()+"    / reverse() = "+a.reverse()+"     / isPalindrome()
= "+a.isPalindrome());
          a = new IntegerNumber(11); System.out.println("  n="+a.getNumber()+"    / reverse() = "+a.reverse()+"    / isPalindrome()
= "+a.isPalindrome());
```

```
          a = new IntegerNumber(19); System.out.println("  n="+a.getNumber()+"    / reverse() = "+a.reverse()+"    / isPalindrome()
= "+a.isPalindrome());
          a = new IntegerNumber(1234); System.out.println("  n="+a.getNumber()+"  / reverse() = "+a.reverse()+"  / isPalindrome() =
"+a.isPalindrome());
          a = new IntegerNumber(4334); System.out.println("  n="+a.getNumber()+"  / reverse() = "+a.reverse()+"  / isPalindrome() =
"+a.isPalindrome());
          a = new IntegerNumber(3434); System.out.println("  n="+a.getNumber()+"  / reverse() = "+a.reverse()+"  / isPalindrome() =
"+a.isPalindrome());
          a = new IntegerNumber(12321); System.out.println("  n="+a.getNumber()+" / reverse() = "+a.reverse()+" / isPalindrome() = "+a.i
          a = new IntegerNumber(12312); System.out.println("  n="+a.getNumber()+" / reverse() = "+a.reverse()+" / isPalindrome() = "+a.i
     }
}
```