differenza_DATE.java

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

    DataOra oggi, nolo;

        oggi = Utility.dtOraSist ();

    String oggi_data, oggi_ora, nolo_data, nolo_ora;

        oggi_data = oggi.getLDateSt();
        oggi_ora  = oggi.getLTimeSt();

        nolo_data = "2024-03-06";
        nolo_ora  = "01:16";

        System.out.println();
        System.out.println("dt-ora nolo : " + nolo_data + " " + nolo_ora);
        System.out.println("dt-ora oggi : " + oggi_data + " " + oggi_ora);

        nolo = new DataOra (nolo_data, nolo_ora);

    long differ = nolo.until(oggi, "MINUTI");
        System.out.println(differ);

    int dif_min, dif_ore, dif_gg;

        dif_min = (int)  differ % 60;
        dif_ore = (int) (differ / 60) % 24;
        dif_gg  = (int) (differ / 60) / 24;
        System.out.println("diffOrario = " + dif_gg + "gg " + dif_ore + "hh " + dif_min + "mi");

        nolo.stampaDati ();
        oggi.stampaDati ();
  }
}
```

Utility.java (aggiungere dopo import java.io.*;**)**

```
import java.time.*;
import java.time.format.*;
import java.time.temporal.ChronoUnit;
class Utility {
 public static DataOra dtOraSist () {
  ZoneId Italia      = ZoneId.of("Europe/Rome");
  ZonedDateTime  adesso;
  LocalDateTime  adessoLDT;
  LocalDate  adessoDate;
  LocalTime  adessoTime;
  DataOra    adessoDtOra;
        adesso        = ZonedDateTime.now( Italia ).truncatedTo(ChronoUnit.MINUTES);
        adessoLDT     = adesso.toLocalDateTime();
        adessoDate    = adesso.toLocalDate();
        adessoTime    = adesso.toLocalTime();
        adessoDtOra= new DataOra (adessoDate, adessoTime);
        return adessoDtOra;
 }
.......
```

DataOra.java

```
import  java.time.*;
import  java.time.format.*;
import  java.time.temporal.ChronoUnit;

class DataOra {
  private LocalDateTime    LDateTime;
  private LocalDate        LDate;
  private LocalTime        LTime;
  private int  aaaa,  mm,  gg;
  private int  ore,    min,  inMin;

  private String     LDateLDTst,     LDateSt,       LTimeSt;

  final private DateTimeFormatter mioFmtTime = DateTimeFormatter.ofPattern("HH:mm");

  public DataOra ( LocalDate LDate , LocalTime LTime ){
    this.LDate  = LDate;
    this.LTime   = LTime;
    LDateTime  = LocalDateTime.of   (LDate , LTime);
    LDateLDTst = LDateTime.format (DateTimeFormatter.ISO_LOCAL_DATE_TIME);
    LDateSt      = LDateTime.format    (DateTimeFormatter.ISO_LOCAL_DATE);
    LTimeSt      = LDateTime.format    (mioFmtTime);

    imposta_attr_int_da_Loc ();

    inMin = min + ore * 60;
  }
  public DataOra ( String LDateSt , String  LTimeSt ){
    this.LDateSt = LDateSt;
    this.LTimeSt = LTimeSt;
    imposta_attr_int_da_Str ();
    inMin = min + ore * 60;
    LDateTime   = LocalDateTime.of (aaaa , mm , gg , ore , min);
    LDate         = LDateTime.toLocalDate();
    LTime         = LDateTime.toLocalTime();
    LDateLDTst  = LDateTime.format (DateTimeFormatter.ISO_LOCAL_DATE_TIME);
  }
  private void imposta_attr_int_da_Loc () {
    aaaa  = LDateTime.getYear();
    mm   = LDateTime.getMonthValue();
    gg     = LDateTime.getDayOfMonth();
    ore    = LDateTime.getHour();
    min   = LDateTime.getMinute();
  }
  private void imposta_attr_int_da_Str  () {
    aaaa = Integer.valueOf( LDateSt.substring(0,4) ).intValue();
    mm  = Integer.valueOf( LDateSt.substring(5,7) ).intValue();
    gg    = Integer.valueOf( LDateSt.substring(8,10) ).intValue();
    ore   = Integer.valueOf( LTimeSt.substring(0,2) ).intValue();
    min  = Integer.valueOf( LTimeSt.substring(3,5) ).intValue();
  }
```

```
  public LocalDateTime getLDateTime () {
    return LDateTime;
  }

  public String getLDateSt () {
    return LDateSt;
  }
  public String getLTimeSt () {
    return LTimeSt;
  }
  public LocalDate getLDate () {
    return LDate;
  }
  public LocalTime getLTime () {
    return LTime;
  }

  public long until ( DataOra fine , String unitaCrono) {
   LocalDateTime fineLDT = fine.getLDateTime();
   long differ;
       if (unitaCrono == "MINUTI")
           differ = this.LDateTime.until (fineLDT, ChronoUnit.MINUTES);
       else
       if (unitaCrono == "SECONDI")
           differ = this.LDateTime.until (fineLDT, ChronoUnit.SECONDS);
       else
           return -999999999;
       return differ;
  }

  public void stampaDati () {
    System.out.println();
    System.out.println("LDateSt   : " + LDateSt + "\t\t LTimeSt : " + LTimeSt);
    System.out.print  ("data      : " + aaaa + " " + mm + " " + gg + "\t\t");
    System.out.println(" ore min :  " + ore  + " " + min + "\t in minuti : " + inMin);

    System.out.println("LDateTime : " + LDateTime  + "\t (oggetto LocalDateTime)");
    System.out.println("LDate     : " + LDate      + "\t\t (oggetto LocalDate)");
    System.out.println("LTime     : " + LTime      + "\t\t (oggetto LocalTime)");
  }
}
```