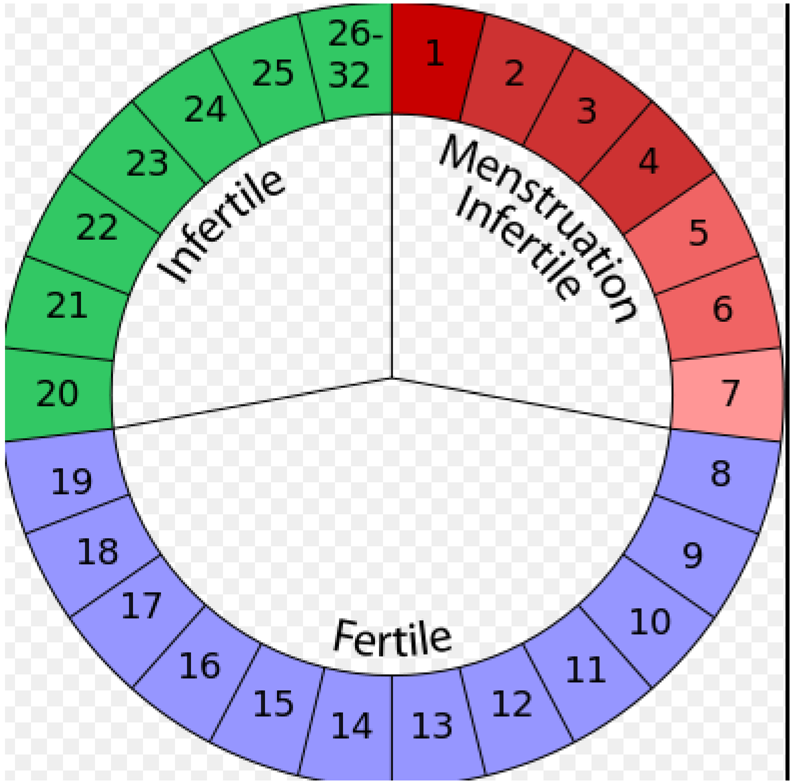
Classe Fertility

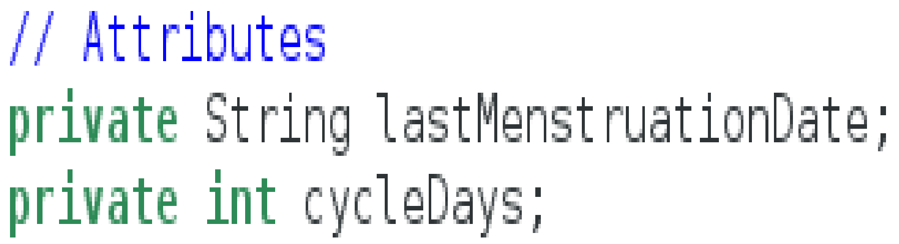


Classe Fertility

- Atributs :
  - Nombre de dies que té el cicle
  - Data de l'última mensturació
    - Cal importar la classe LocalDateTime

      `import java.time.LocalDateTime;`

Classe Fertility



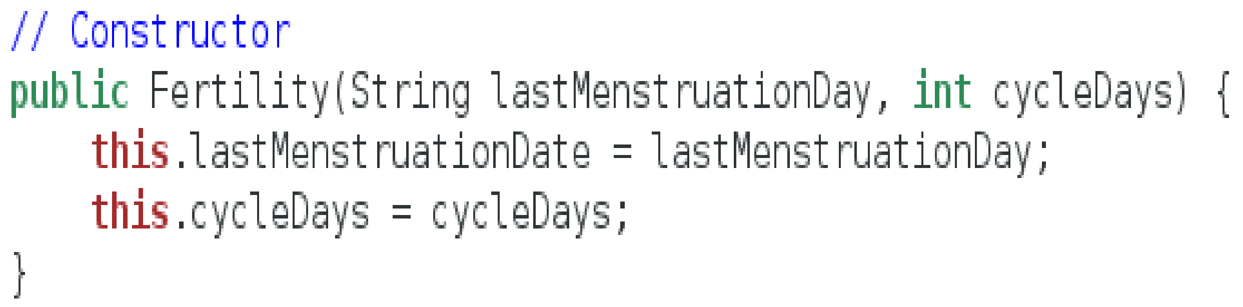
```
// Attributes
private String lastMenstruationDate;
private int cycleDays;
```

Classe Fertility

- Constructor

?

Classe Fertility

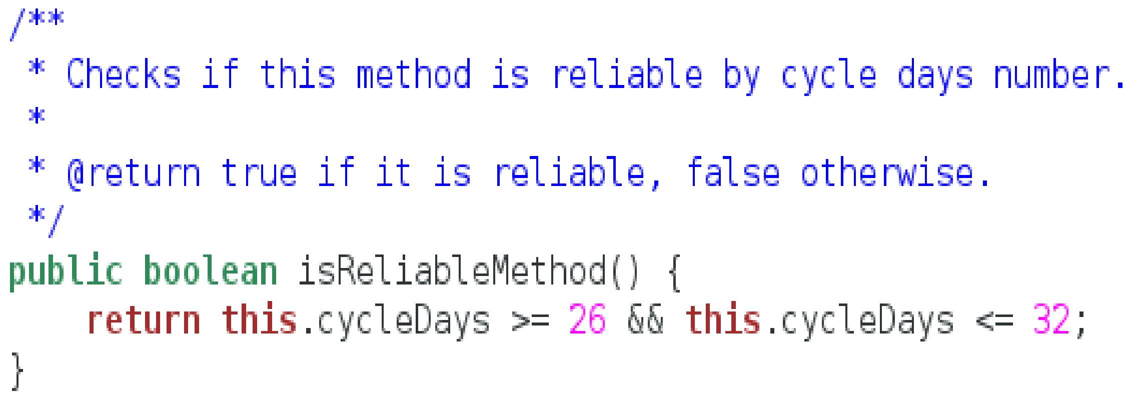


```java
// Constructor
public Fertility(String lastMenstruationDay, int cycleDays) {
    this.lastMenstruationDate = lastMenstruationDay;
    this.cycleDays = cycleDays;
}
```

Classe DateTime

- Mètodes :
  - public LocalDateTime minusDays(int days)
    - Retorna el nombre de dies entre dues dates
  - public static LocalDateTime now()
    - Retorna la data actual del sistema
  - public boolean isAfter(long instant) (metode heretat de la classe AbstractInstant
    - Compara si una data és posterior a una altre
  - public boolean isBefore(long instant) (metode heretat de la classe AbstractInstant
    - Compara si una data és anterior a una altre

Classe Fertility

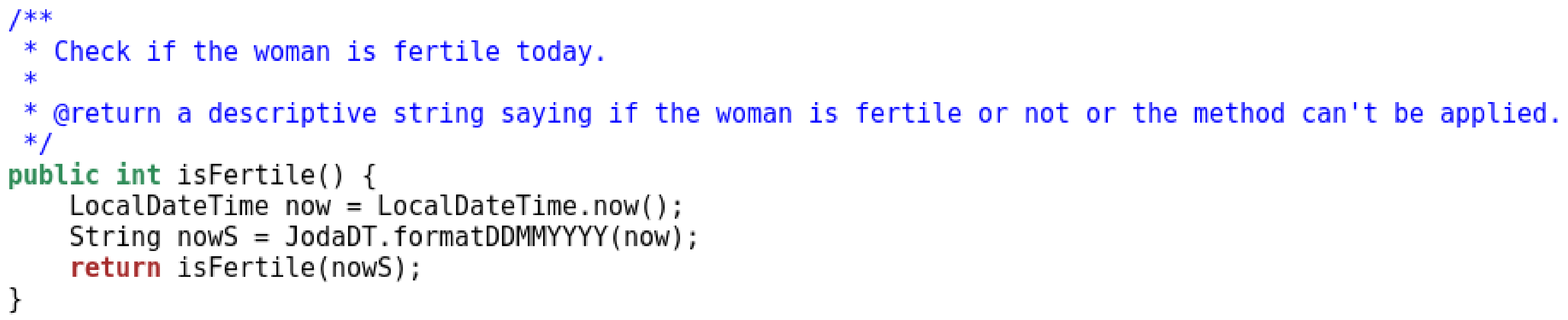


```
/**
 * Checks if this method is reliable by cycle days number.
 *
 * @return true if it is reliable, false otherwise.
 */
public boolean isReliableMethod() {
    return this.cycleDays >= 26 && this.cycleDays <= 32;
}
```

Classe Fertility

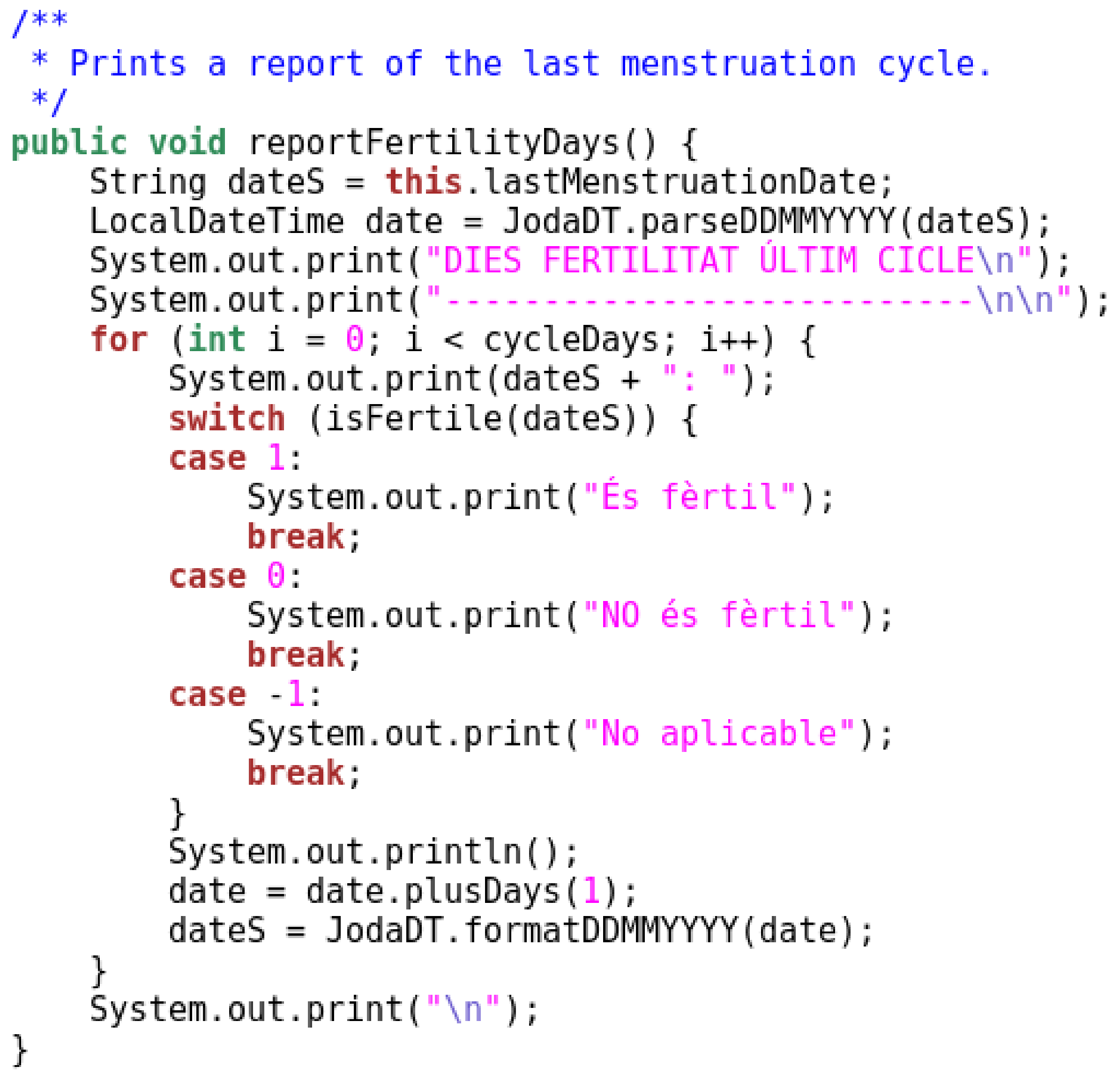
```java
/**
 * Check if the woman is fertile on a date.
 *
 * @param date a date
 * @return 1 if the woman is fertile, 0 if the woman is not fertile and -1 if the method can't be applied.
 */
public int isFertile(String date) {
    LocalDateTime dt = JodaDT.parseDDMMYYYY(date);
    LocalDateTime dtl = JodaDT.parseDDMMYYYY(this.lastMenstruationDate);
    // Between the 1rst and the 7th days, she is not fertile
    if (dt.isAfter(dtl.minusDays(1)) && dt.isBefore(dtl.plusDays(7))) {
        return 0;
    }
    // Between the 8th and the 19th day, she is fertile
    if (dt.isAfter(dtl.plusDays(6)) && dt.isBefore(dtl.plusDays(19))) {
        return 1;
    }
    // Between the 20th and the last day, she is not fertile
    if (dt.isAfter(dtl.plusDays(18)) && dt.isBefore(dtl.plusDays(this.cycleDays))) {
        return 0;
    }
    // it cannot be applied
    return -1;
}
```

Classe Fertility



```java
/**
 * Check if the woman is fertile today.
 *
 * @return a descriptive string saying if the woman is fertile or not or the method can't be applied.
 */
public int isFertile() {
    LocalDateTime now = LocalDateTime.now();
    String nowS = JodaDT.formatDDMMYYYY(now);
    return isFertile(nowS);
}
```

Classe Fertility



```java
/**
 * Prints a report of the last menstruation cycle.
 */
public void reportFertilityDays() {
    String dateS = this.lastMenstruationDate;
    LocalDateTime date = JodaDT.parseDDMMYYYY(dateS);
    System.out.print("DIES FERTILITAT ÚLTIM CICLE\n");
    System.out.print("---------------------------\n\n");
    for (int i = 0; i < cycleDays; i++) {
        System.out.print(dateS + ": ");
        switch (isFertile(dateS)) {
        case 1:
            System.out.print("És fèrtil");
            break;
        case 0:
            System.out.print("NO és fèrtil");
            break;
        case -1:
            System.out.print("No aplicable");
            break;
        }
        System.out.println();
        date = date.plusDays(1);
        dateS = JodaDT.formatDDMMYYYY(date);
    }
    System.out.print("\n");
}
```