

ÍNDEX

ACTIVITAT 6.13 Representació i maneig de dates (Classe Data)	1
Enunciat	1
Atributs	2
Constructors	2
Mètodes públics d'objecte	2
Mètodes estàtics públics	3
Mètodes privats d'objecte	4

ACTIVITAT 6.13 Representació i maneig de dates (Classe Data)



Abans de començar:

Descàrrega d'este [enllaç](#) la **plantilla de la classe Data** què hauràs d'anar completant.

A més de la plantilla de la classe **Data** (fitxer *Data.java*, dins del paquet *main*), també disposes d'una classe **TestFecha** (fitxer *TestFecha.java*, dins del paquet *test*) que conté *proves unitàries* (tests) dels mètodes públics que es demanen implementar en esta activitat.

Executa aquestos tests per a comprovar si la implementació que vas realitzant en la plantilla és correcta per a passar (color verd) estos tests. No obstant això, hauràs de realitzar proves addicionals per tal d'assegurar que els teus mètodes estàn correctament implementats tal com es descriu en el següent enunciat.

Enunciat



La representació i maneig de dates en una aplicació és una **tasca complexa**. En esta activitat crearem la nostra pròpia classe que inclourem en altres aplicacions pròpies que realitzarem amb posterioritat.

La classe **Data** haurà de complir les següents especificacions (es recomana seguir l'[ordre de implementació](#) proposat per a dur a

terme la práctica amb éxit):

Atributs

- La data vindrà representada per **3 atributs de tipus sencer** corresponents al dia, *mes i any* de la data representada per l'objecte

Constructors

- Constructor **per defecte** (sense paràmetres) que iniciarà la data a l'1 de gener de l'any 1970.
- Constructor amb **3 paràmetres sencers** que correspondrà al dia, mes i any.
- Constructor amb **1 paràmetre de tipus String** que rebrà una data en format **dd/mm/yyyy**. Haurà de trossejar i convertir la cadena per a inicialitzar l'objecte. Has d'utilitzar les funcions de `String` o consultar la documentació oficial i fer-ho OBLIGATÒRIAMENT amb la classe [StringTokenizer](#). Si el format rebut no és el correcte, crearà la data per defecte.

Mètodes públics d'objecte

- **Mètodes consultors (getters)** per als atributs dia, mes i any.
- `public void set(int dia, int mes, int any)` Mètode **modificador parametritzat**; modifica la data de l'objecte actual.
- `public Data clone()`: Crearà i retornarà un nou objecte de la classe `Data` iniciat amb els valors que representa l'objecte actual, és a dir, crearà un clon exacte de l'objecte actual.
- `public void mostrarEnFormatES()`: Mostrarà la data per pantalla en format espanyol dd-mm-yyyy. **Ex. 14-08-2020**
- `public void mostrarEnFormatGB()`: Mostrarà la data per pantalla en format anglés yyyy-mm-dd. **Ex. 2020-08-14**
- `public void mostrarEnFormatText()`: Mostrarà la data per pantalla en format Text: **Ex. 14-agost-2020**.
- `public boolean isIgual(Data data)`: Retorna **true/false** indicant si la classe de l'objecte actual representa la mateixa data que aquella representada per l'objecte que rep com a paràmetre. Haurem de comprovar un a un si els atributs són iguals.
- `public String getDiaSetmana()`: Retorna un `String` que representa el dia de la setmana en format text (**dilluns, dimarts, dimecres, dijous, divendres, dissabte, diumenge**). L'algorisme per a calcular el dia de la setmana és el següent.
 1. Fent ús del mètode estàtic [getDiesTranscorregutsOrigen](#) definit posteriorment, calcula el nombre de dies que han passat des de l'origen, és a dir, des de la data 1-1-1.
 2. Calcula la resta de la divisió entre 7 de la quantitat de dies que han passat des de l'origen.

3. El valor obtingut serà el dia de la setmana de la data actual:
0→ diumenge, 1-> dilluns, 2→ dimarts, 3→ dimecres,...

- `public boolean isFestiu()`: Retorna un booleà indicant si la data representada per l'objecte actual és festiva. **Es considerarà festiu si el dia de la setmana és dissabte o diumenge.**
- `public int getNumeroSetmana()`: Retorna el número de la setmana dins de l'any actual. Es considera una setmana l'interval de dates entre una data que siga dilluns i la següent data en ordre cronològic que siga diumenge. També es comptabilitza com a setmana tant la primera setmana de l'any com l'última (inclusivament en aquells anys en què la primera i/o última setmana no conté set dies en total).
- `public Data afegir(int dies)`: Retorna un **nou objecte de tipus data** que representa la data resultant d'afegir el nombre de dies passats com a argument a la data que representa l'objecte actual. **Haurem de tindre en compte els dies que té el mes actual i si l'any és de traspàs (bisiesto) amb la finalitat de construir el nou objecte amb la data correcta.** El màxim nombre de dies que podem afegir serà 30 i no podem afegir un nombre negatiu de dies.
- `public Data restar(int dies)`: Retorna un **nou objecte de tipus data** que representa la data resultant de restar el nombre de dies passats com a argument a la data que representa l'objecte actual. **Haurem de tindre en compte els dies que té el mes actual i si l'any és de traspàs (bisiesto) amb la finalitat de construir el nou objecte amb la data correcta.** El màxim nombre de dies que podem restar serà 30 i no podem restar un nombre negatiu de dies.
- `public boolean isCorrecta()`: Retorna un booleà indicant si la data representada per l'objecte actual és correcta. No oblidis comprovar que el dia es trobe dins del rang dels dies que té el mes tenint en compte si l'any és de traspàs(bisiesto) o no.

Mètodes privats d'objecte

- Crea tots els **mètodes privats d'objecte** que consideres necessaris per a dur a terme les funcions exposades pels mètodes públics. Es proposen definir els següents però com és la **part privada** de la classe pots decidir realitzar-ho com més t'agrade, encara que es recomanable seguir aquesta proposta:
 - `private String getMesEnFormatText()`: Retorna el mes representat per la classe actual en format text (gener, febrer, març,...)
 - `private int getDiesTranscorregutsEnAny()`: Retorna el nombre de dies que han passat des del dia 1 de l'any en curs.
 - `private int getDiesTranscorregutsOrigem()`: Retorna el nombre de dies transcorreguts des de l'origen 1-1-1. Estos dies vindran donats pels dies que tenen cadascun dels anys anteriors i els dies que han passat en any en curs. Hauràs de fer ús dels mètodes estàtics [getDiesAny](#) i [getDiesMes](#) definits en la secció posterior.

Mètodes estàtics públics

Has d'implementar obligatòriament els següents mètodes estàtics (de classe). També pots implementar addicionalment els mètodes privats que cregues necessaris:

- `public static boolean isBisiesto (int any)`: Retorna un booleà indicant si l'any passat com a argument és de traspàs. Un **any és de traspàs si és divisible per 4 però no és divisible entre 100 o és divisible entre 4 entre 100 i entre 400**.
- `public static int getDiesAny (int any)`: fent ús de la funció **isBisiesto**, retorna el nombre de dies que té l'any passat com a argument.
- `public static int getDiesMes (int mes, int any)`: Fent ús de la funció **isBisiesto**, retorna el nombre de dies que té el mes passat com a argument en un any determinat (passat també com a argument).

Ordre recomanat de implementació dels diferents mètodes

1. getters
2. constructors
3. set
4. clone
5. mostrarEnFormatXXX
6. isIgual
7. isBisiesto
8. getDiesAny
9. getDiesMes
10. getMesEnFormatText
11. getDiesTranscorregutsEnAny
12. getDiesTranscorregutsOrigen
13. getDiaSetmana
14. isFestiu
15. getNumeroSetmana
16. afegir i restar
17. isCorrecta

Per a finalitzar, crea també una classe **TestData** (dins del mateix paquet on es troba la classe **Data**) en la qual realitzarem proves addicionals per a comprovar la correcta implementació de tots els constructors i mètodes de la classe que acabes de crear.

A continuació, es presenta un exemple d'execució, encara que has de realitzar altres proves addicionals o alternatives ja que en l'exemple d'execució no es prova tots els mètodes plantejats:

Exemples d'execució TestFecha

```
=== START - Prova de Constructors - START ===  
  
--- Cree un nou objecte utilitzant el constructor parametritzat int int  
int (16,1,2021) ---  
16-01-2021  
2021-01-16  
16-gener-2021
```

```
La data és correcta: true
La data és festiu: true
El dia de la setmana és: dissabte
-----

--- Cree un nou objecte utilitzant el constructor String 16/1/2021 ---
16-01-2021
2021-01-16
16-gener-2021
La data és correcta: true
La data és festiu: true
El dia de la setmana és: dissabte
-----

--- Comprove mitjançant el mètode isIgual que les dates representades
per tots dos objectes són iguals -

La data creada amb els constructors anteriors són iguals: true
-----

--- Cree un nou objecte utilitzant el constructor per defecte ---
01-01-1970
1970-01-01
01-gener-1970
La data és correcta: true
La data és festiu: false
El dia de la setmana és: dijous
-----

=== FI - Prova de Constructors - FI ===

=== START - Prova de Mètodes afegir/restar dies - START ===

--- Endemà de la data inicial (16-1-2021) - (+1 dia) ---
17-01-2021
2021-01-17
17-gener-2021
La data és correcta: true
La data és festiu: true
El dia de la setmana és: diumenge
-----

--- Dia anterior a la data inicial (16-1-2021) - (-1 dia) ---
15-01-2021
2021-01-15
15-gener-2021
La data és correcta: true
La data és festiu: false
El dia de la setmana és: divendres
-----
```

```
--- Data corresponent a restar 30 dies a la data inicial (16-1-2021) -  
(-30 dies) ---  
17-12-2020  
2020-12-17  
17-deseembre-2020  
La data és correcta: true  
La data és festiu: false  
El dia de la setmana és: dijous  
-----  
  
=== FI - Prova de Mètodes afegir/restar - FI ===  
  
=== START - Prova del mètode modificador - START ===  
  
--- Modifique la data del primer objecte creat (16-1-2020) per la data  
22-1-2021 ---  
22-01-2021  
2021-01-22  
22-gener-2021  
La data és correcta: true  
La data és festiu: false  
El dia de la setmana és: divendres  
-----  
  
=== FI - Prova del mètode modificador - FI ===
```