

## Unitat Formativa 5. POO. Llibreries de classes fonamentals

### Activitat 2. Tractament de dates i tractament d'excepcions

**Objectiu:** Treballar amb classes que contenen atributs que també són classes i que ens ofereixen majors possibilitats als nostres serveis, tractant les errades.

#### Completar els següents apartats:

- Crear al package **exceptions** una excepció **DateException** que hereti de **Exception**, per utilitzar-la amb els següents mètodes **set** i validar que els strings rebuts que contenen dates són vàlids, llençant una **DateException** en cas contrari.

- A la classe **Checker** del Package **model/check** crear un **mètode estàtic** per validar qualsevol data rebuda com string, com el següent:

```
public static function checkDate (string $data) { //definir funcionalitat }
```

Aquesta funció llençarà la excepció **DateException** quan la data rebuda com string no sigui vàlida per ser convertida a un objecte de classe **DateTime**.

- Modificar les classes on tenim atributs de classe **DateTime** per utilitzar el mètode **checkDate** al seu corresponent mètode set.
- Modificar la classe **Employee** per afegir el següent mètode:
  - **getDays():int** que retorni el nombre de dies que resta contractat l'empleat.
- Modificar la classe **Book** per treballar amb els 2 atributs de classe **DateTime** (la data de publicació del llibre -sense informació horària- i la data on el vam tenir disponible –amb informació horària-). Afegir els mètodes:
  - **getPeriod():string**, on retornem un string que conté el temps transcorregut entre la seva publicació i quan l'hem tingut disponible amb format "**X dies, Y mesos, Z anys**".
  - **getIntervals(int \$days):array**, on retornem un array que conté les dates transcorregudes entre la data de disponibilitat al nostre negoci i la data actual, amb l'interval de dies entre dates del array igual al valor rebut per paràmetre.

- **Comprovar tot el codi afegit amb els tests funcionals.**