

## ACTIVITAT 9.2 Matriu genèrica

Temporalització: SENSE DEFINIR

Agrupament: INDIVIDUAL

### Què necessites per a fer esta activitat?

✓ Entorn de desenvolupament Netbeans

### Enunciat

Descarrega el projecte MatriuGenerica.zip d'AULES i importa'l a Netbeans

Has d'analitzar el codi del fitxer MatriuGenerica.java, el qual implementa la gestió de suma i multiplicació de matrius de nombres enters. Fent una ullada al codi d'esta classe, veuràs que la classe disposa de tres mètodes públics:

```
public void sumar (Integer[][] matriu)

public void multiplicar (Integer[][]matriz)

public void imprimirUltimaOperacio()
```

A més, pots vore que la classe disposa de 4 atributs privats:

```
private Integer[][] matriuOpEsquerra;
```

- Emmagatzema la matriu operant esquerra de l'última operació realitzada

```
private Integer[][] matrizOpDreta;
```

- Emmagatzema la matriu operant dreta de l'última operació realitzada

```
private TipusOperacio tipusOperacio;
```

- Emmagatzema el tipus d'operació de l'última operació realitzada (ja siga una suma o una multiplicació)

```
private Integer[][] matriu;
```

- Emmagatzema la matriu resultat de l'última operació realitzada

Una volta analitzat el codi, has de fer els canvis següents:

1. **Transforma el codi MatriuGenerica a una classe abstracta i genèrica** per a que esta classe permeta treballar amb qualsevol tipus de dada que herete de la classe Number de Java.
2. A més dels mètodes ja existents, la classe MatriuGenerica tindrà quatre mètodes abstractes anomenats **suma**, **multiplicació**, **zero** i **aleatori**. Els quatre mètodes retornen una dada del tipus genèric escollit. El mètode "suma" i el de "multiplicació" rebran dos paràmetres que seran del tipus de dada genèrica amb què treballa la matriu.

Dels mètodes existents has de substituir l'operació de suma, multiplicació i valor 0 per la invocació a estos mètodes abstractes.

Per exemple substituir: `resultat[i][j] = this.matriu[i][j] + matriu[i][j];`

Per: `resultat[i][j] = this.suma(this.matriu[i][j], matriu[i][j]);`

3. Implementa ara una classe MatriuEnters que heretarà de MatriuGenerica. MatriuEnters concretarà el tipus de dada de la MatriuGenerica i implementarà els mètodes suma, multiplicació i zero per a este tipus de matrius concretes.
4. Realitza una nova classe TestMatriuEnters per a provar els mètodes de la nova classe creada. Prova almenys a fer una suma i una multiplicació i mostrar els resultats.
5. Crea ara una classe Fraccio que heretarà de la classe Number i que tindrà dos atributs privats numerador i denominador. Esta classe disposarà també de dos mètodes públics **suma** i **multiplica** que sumarà i multiplicarà a la fracció una nova fracció que es proporcionarà com a paràmetre i retornarà la fracció resultant.
6. Implementa una nova classe MatriuFraccions que permeti fer la suma i la multiplicació de matrius de fraccions.
7. Realitza una classe TestMatriuFraccions on proves la classe MatriuFraccions, realitzant una prova similar a la que has realitzat amb la matriu d'enters.