


La généricité

En Programmation Orientée Object (POO), la généricité est un concept permettant de définir des algorithmes (types de données et méthodes) identiques qui peuvent être utilisés sur de multiples types de données.

```
public class Account {  
  
    private String customer;  
    private double amount;  
    private char currency;  
  
    public Account(String customer, double amount, char currency)  
    {  
        this.setCustomer = customer;  
        this.setAmount = amount;  
        this.setCurrency = currency;  
    }  
  
    {...getters & setters... }  
}
```

```
public class App {  
  
    public static void main(String[] args)  
    {  
        Account myAccount = new Account("Thor", 1000, '€');  
  
        Account otherAccount = new Account("Groot", 100, "euros");  
    }  
}
```

Euh ! Non ! moi j'attend un caractère et sûrement pas une chaîne de caractère !



On passe à notre class le type **générique**,
Puis on l'attribut comme type à une
variable

```
public class Account<T> {  
  
    private String customer;  
    private double amount;  
    private T currency;  
  
    public Account(String customer, double amount, T currency)  
    {  
        this.setCustomer = customer;  
        this.setAmount = amount;  
        this.setCurrency = currency;  
    }  
  
    {...getters & setters... }  
}
```

```
public class App {  
  
    public static void main(String[] args)  
    {  
        Account<Character> myAccount = new Account<Character>("Thor", 1000, '€');  
  
        Account<String> otherAccount = new Account<String>("Groot", 100, "euros");  
    }  
}
```

Au moment de l'**instanciation**, on précise
le type qui fera référence au **type**
générique, en l'occurrence **T** dans
l'exemple

Plusieurs type générique sont disponibles :

