

Notions

- ▶ **Héritage** : l'héritage permet de regrouper des informations communes de plusieurs classes pour définir une structure « super globale ».
- ▶ **Protected** : un attribut ou une fonction « protected » est un élément défini dans une classe mère accessible depuis une classe fille, mais inaccessible depuis un algorithme tiers
- ▶ **Parent** : Le mot clef « parent » permet d'accéder aux informations de la classe mère
- ▶ **Abstract** : une fonction abstraite est une fonction qui ne dispose pas de contenu et qui doit être obligatoirement implémentée par les classe filles. Une classe qui contient une fonction abstraite est forcément abstraite elle-même et ne peut pas être instanciée.
- ▶ **Interface** : une interface est une classe totalement abstraite. Elle permet de forcer l'implémentation de fonctions dans une classe qui l'utilise.

Exemples

```
INTERFACE Combat
    PUBLIC FONCTION infligerDegat();
    PUBLIC FONCTION pertePV(pv)
FINCLASSE
```

```
ABSTRACT CLASSE Personnage implements Combat
    //contenu
FINCLASSE
```

```
CLASSE Humain extends Personnage{
    //contenu normal
    PUBLIC FONCTION infligerDegat(){
        RETOURNER this.calculDegat() / 2
    }
    PUBLIC FONCTION pertePV(pv){
        this.pv -= 10;
    }
}
```

```
CLASSE Zombie extends Personnage{
    //contenu normal
    PUBLIC FONCTION infligerDegat(){
        RETOURNER this.calculDegat() * 2
    }
    PUBLIC FONCTION pertePV(pv){
        this.pv -= pv;
    }
}
```