

Programació II

2. Programació Orientada a Objectes

2.1 Polimorfisme

Exercici – Pokemons



- 1) Implementa la classe *Pokemon* amb els atributs (o membres de dades) protegits *name* i *type*., i els següents membres funció (o mètodes) públics:
 - a. Un constructor que rebi un string amb el que inicialitzi *name* i un tipus per *type*, amb un valor per defecte “noType”
 - b. *getName()* que retorni el nom del Pokemon
 - c. *getType()* que retorni el tipus del Pokemon
- 2) Implementa les classes *ElectricPokemon*, *GrassPokemon*, *FirePokemon* i *WaterPokemon* derivades de *Pokemon*. Cadascuna d’elles ha de proveir els següents mètodes (o membres funció) públics:
 - a. Un constructor que rebi un string pel nom del Pokemon i inicialitzi l’atribut heretat *type* amb el tipus específic de cada classe: “Electric”, “Grass”, “Fire” i “Water” respectivament.
 - b. Un mètode *damageAttack(...)* amb el que donat un apuntador a un pokemon retorni el dany (un enter) que el pokemon infringeix a un altre pokemon (el que se li passa com a paràmetre).

Per això, considera que el dany per defecte és 10 unitats i depenent del tipus de pokemon atacat (el paràmetre) aquest valor es modifica com s’indica en aquesta taula:

	Electric	Grass	Fire	Water
Electric		½		2
Grass			½	2
Fire		2		½
Water		½	2	

Ex. Un pokemon de tipus electric dobla el dany del seu atac contra un pokemon de tipus water, mentre que el dany produït a un pokemon de tipus grass és la meitat.

Donat el següent codi a la funció main:

```
int main()
{
    Pokemon* pikachu = new ElectricPokemon("Pikachu");
    Pokemon* bulbasaur = new GrassPokemon("Bulbasaur");
    Pokemon* charmander = new FirePokemon("Charmander");
    Pokemon* squirtle = new WaterPokemon("Squirtle");

    std::cout << "-- Damage Attacks --" << std::endl;

    std::cout << pikachu->getName() << " -> " << bulbasaur->getName()
              << " = " << pikachu->damageAttack(bulbasaur) << " units\n";

    std::cout << bulbasaur->getName() << " -> " << squirtle->getName()
              << " = " << bulbasaur->damageAttack(squirtle) << " units\n";

    std::cout << squirtle->getName() << " -> " << charmander->getName()
              << " = " << squirtle->damageAttack(charmander) << " units\n";

    std::cout << charmander->getName() << " -> " << pikachu->getName()
              << " = " << charmander->damageAttack(pikachu) << " units\n\n";

    delete pikachu;
    delete bulbasaur;
    delete charmander;
    delete squirtle;

    system("pause");
    return 0;
}
```

L'aplicació ha de generar la següent sortida:

```
-- Damage Attacks --
Pikachu -> Bulbasaur = 5 units
Bulbasaur -> Squirtle = 20 units
Squirtle -> Charmander = 20 units
Charmander -> Pikachu = 10 units
```