

Reporte de la Tarea 1

Luis Diego Fernández Coto - B22492

20 de enero de 2017

Índice

1. Enunciado	2
2. Código	4
2.1. main.cpp	4
2.2. Pokemon.h	5
2.3. Pokemon.cpp	5
2.4. Flying.h	6
2.5. Flying.cpp	6
2.6. Ice.h	7
2.7. Ice.cpp	7
2.8. Electric.h	8
2.9. Electric.cpp	8
2.10. Fire.h	9
2.11. Fire.cpp	9
2.12. Articuno.h	10
2.13. Articuno.cpp	10
2.14. Zapdos.h	12
2.15. Zapdos.cpp	13
2.16. Moltres.h	14
2.17. Moltress.cpp	15
3. Conclusiones	18

1. Enunciado

IE-0217 Estructuras abstractas de datos y algoritmos para ingeniería

Herencia en Pokémon: Articulo, Moltress, Zapdos

M. Sc. Ricardo Román Brenes - ricardo.roman@ucr.ac.cr

III-2016

Tabla de contenidos

1. Enunciado	1
2. Consideraciones	2

1. Enunciado

Implemente un programa en C++ utilizando herencia que modele las tres aves legendarias de Pokémon: Articulo, Moltress y Zapdos:

- Implemente una clase abstracta Pokemon que contenga:
 - Atributos: name, species, HP, ATK, DEF, sATK, sDEF, SPD, EXP, call.
 - Métodos:
 - virtual void atk1(Pokemon &other) = 0
 - virtual void atk2(Pokemon &other) = 0
 - virtual void atk3(Pokemon &other) = 0
 - virtual void atk4(Pokemon &other) = 0
 - string call()
 - void printInfo()
- Implemente cuatro clases llamadas Electric, Fire, Water y Flying que modelan los tipos de las aves míticas. Estas clases heredan de la clase Pokémon.
- Métodos:
 - static string type()

- static string strongVs()
- static string weakVs()
- Implemente tres clases concretas llamadas Zapdos, Articulo y Moltress.
- Atributos
 - Type, strongVs, weakVs.
- Métodos
 - void print()

Las clases Zapdos, Articulo y Moltress deben implementar los métodos virtuales puros (¡sea creativo!) y un método que imprima el estado del Pokemon con todos sus atributos.

Haga un programa de prueba en donde se creen dos objetos e interactúen entre ellos (se ataquen un par de veces).

Escriba también un Makefile con al menos tres reglas:

1. build: para compilar su programa.
2. run: para ejecutar el programa de prueba.
3. clean: para eliminar ejecutables y archivos intermedios.

2. Consideraciones

- Trabajo individual.
- Genere un reporte en \LaTeX que incluya el enunciado, su código y sus conclusiones.
- Suba su código con documentación interna al GitHub respectivo de su grupo y el directorio del laboratorio.
- Cada estudiante debe subir el reporte a Schoology. (<https://app.schoology.com/assignment/939704424/>).
- Recuerde que por cada día tardío de entrega se le rebajaran puntos de acuerdo con la formula: 3^d , donde $d > 1$ es la cantidad de días naturales tardíos.

2. Código

2.1. main.cpp

```
#include <iostream>

#include "Articuno.h"
#include "Electric.h"
#include "Fire.h"
#include "Flying.h"
#include "Ice.h"
#include "Moltres.h"
#include "Pokemon.h"
#include "Zapdos.h"

using namespace std;

int main()
{
    //se crean 3 objetos que heredan de pokemon, pero no son iguales
    Zapdos* p1 = new Zapdos("Piolin");
    Articuno* p2 = new Articuno("Pinguino");
    Moltres* p3 = new Moltres("Yiguirro");

    //se imprime cada objeto para ver sus atributos
    cout << "Estado de los 3 pokemones creados\n" << endl;
    p3->print();
    p2->print();
    p1->print();

    //se procede a utilizar las funciones de los ataques de cada pakemon
    //despues de cada ataque se muestra el estado de los pokemones afectados
    cout << "Fase de ataques\n" << endl;
    p2->atk1(*p1);
    p1->print();

    p2->atk3();
    p2->print();

    p3->atk4(*p2);
    p2->print();
    p3->print();

    //se procede a liberar la memoria del Heap section utilizada al hacer los new
    delete p1;
    delete p2;
    delete p3;
```

```

    return 0;
}

```

2.2. Pokemon.h

```

#ifndef POKEMONH
#define POKEMONH

#include <iostream>

using namespace std;

class Pokemon {

//atributos de la clase pokemon
public:
    string name;
    string species;
    string cry;
    string status;
    int SPD;
    int HP;
    int ATK;
    int DEF;
    int sATK;
    int sDEF;
    int EXP;

public:
    Pokemon();
    virtual ~Pokemon();

    void printInfo ();
    string call ();
    //definicion de 4 metodos virtuales puros
    //no se implementan
    virtual void atk1(Pokemon &other) = 0;
    virtual void atk2(Pokemon &other) = 0;
    virtual void atk3() = 0; //este ataque tiene implicaciones sobre el mismo pokemon
                             // se implemento asi y no como dice el enunciado de la
    virtual void atk4(Pokemon &other) = 0;

};

#endif // POKEMONH

```

2.3. Pokemon.cpp

```

#include "Pokemon.h"

```

```

Pokemon::Pokemon() {

}

Pokemon::~~Pokemon() {

}

string Pokemon::call() {
    return cry;
}

//imprime informacion basica general de cualquier pokemon
void Pokemon::printInfo() {
    cout << "Nombre: " << name << endl;
    cout << "Especie: " << species << endl;
    cout << "Estado: " << status << endl;
    cout << "Speed: " << SPD << endl;
    cout << "Hp: " << HP << endl;
    cout << "Special attack: " << sATK << endl;
    cout << "Special defense: " << sDEF << endl;
    cout << "Experience: " << EXP << endl;
}

```

2.4. Flying.h

```

#ifndef FLYING_H
#define FLYING_H

#include "Pokemon.h"

using namespace std;

//virtual public Pokemon nos dice que hereda de pokemon
class Flying : virtual public Pokemon {
public:
    Flying();
    virtual ~Flying();

    static string type();
    static string strongVs();
    static string weakVs();
};

#endif // FLYING_H

```

2.5. Flying.cpp

```

#include "Flying.h"

Flying::Flying() {

}

Flying::~~Flying() {

}

string Flying::type () {
    return "Flying ";
}

string Flying::strongVs () {
    return "Grass Fight Bug ";
}

string Flying::weakVs () {
    return "Electric Ice Rock ";
}

```

2.6. Ice.h

```

#ifndef ICE_H
#define ICE_H

#include "Pokemon.h"

using namespace std;

//virtual public Pokemon nos dice que hereda de pokemon
class Ice : virtual public Pokemon {
public:
    Ice();
    virtual ~Ice();

    static string type();
    static string strongVs();
    static string weakVs();
};

#endif // ICE_H

```

2.7. Ice.cpp

```

#include "Ice.h"

Ice::Ice()

```

```

{

}

Ice::~Ice() {

}

string Ice::type () {
    return "Ice ";
}

string Ice::strongVs () {
    return "Grass Ground Flying Dragon ";
}

string Ice::weakVs () {
    return "Fire Fight Rock Steel ";
}

```

2.8. Electric.h

```

#ifndef ELECTRIC_H
#define ELECTRIC_H

#include "Pokemon.h"

using namespace std;

//virtual public Pokemon nos dice que hereda de pokemon
class Electric : virtual public Pokemon {
public:
    Electric();
    virtual ~Electric();

    static string type();
    static string strongVs();
    static string weakVs();
};

#endif // ELECTRIC_H

```

2.9. Electric.cpp

```

#include "Electric.h"

Electric::Electric() {

}

```



```

Electric::~Electric() {

}

string Electric::type () {
    return "Electric ";
}

string Electric::strongVs () {
    return "Water Flying ";
}

string Electric::weakVs () {
    return "Ground ";
}

```

2.10. Fire.h

```

#ifndef FIRE_H
#define FIRE_H

#include "Pokemon.h"

using namespace std;

//virtual public Pokemon nos dice que hereda de pokemon
class Fire : virtual public Pokemon {
public:
    Fire();
    virtual ~Fire();

    static string type();
    static string strongVs();
    static string weakVs();
};

#endif // FIRE_H

```

2.11. Fire.cpp

```

#include "Fire.h"

Fire::Fire() {

}

Fire::~~Fire () {

```

```

}

string Fire::type () {
    return "Fire ";
}

string Fire::strongVs () {
    return "Grass Ice Bug Steel ";
}

string Fire::weakVs () {
    return "Water Ground Rock ";
}

```

2.12. Articuno.h

```

#ifndef ARTICUNO_H
#define ARTICUNO_H

#include "Flying.h"
#include "Ice.h"

using namespace std;

//no dice que esta clase hereda de Ice y de Flying
class Articuno : public Ice, public Flying {
public:
    string type;
    string strongVs;
    string weakVs;

public:
    Articuno ();
    Articuno(string nombre);
    virtual ~Articuno();

    void atk1 (Pokemon &other);
    void atk2 (Pokemon &other);
    void atk3 ();
    void atk4 (Pokemon &other);
    void print();

};

#endif // ARTICUNO_H

```

2.13. Articuno.cpp

```

#include "Articuno.h"

```

```

Articuno::Articuno() {

}

//implementacion del constructor del constructor
Articuno::Articuno(string nombre) {
    this->name = nombre;
    this->species = "Articuno";
    this->type = Ice::type() + Flying::type();
    this->cry = "prrrnnniinnprnnn";
    this->status = "Healthy";
    this->SPD = 85;
    this->HP = 90;
    this->ATK = 85;
    this->DEF = 100;
    this->sATK = 95;
    this->sDEF = 125;
    this->EXP = 0;
    this->strongVs = Ice::strongVs() + Flying::strongVs();
    this->weakVs = Ice::weakVs() + Flying::weakVs();
}

Articuno::~~Articuno() {

}

void Articuno::print() {
    printInfo();
    cout << "Strong against: " << strongVs << endl;
    cout << "Weak against: " << weakVs << endl;
    cout << "\n";
}

//Freeze dry
void Articuno::atk1(Pokemon &other) {
    cout << this->name << " attacks " << other.name << " with Freezze dry " << endl;
    cout << "Enemy got frozen! " << "\n" << endl;
    other.status = "Frozen";
    other.SPD = 0;
}

//Ice Beam
void Articuno::atk2(Pokemon &other) {
    cout << this->name << " attacks" << other.name << " with Ice Beam " << "\n" << endl;
    if (other.HP <= 90) {
        other.HP = 0;
        if (EXP + 75 < 100)

```

```

        EXP += 75;
    else
        EXP = (EXP + 75 - 100);
    }
    else
        other.HP -= 90;
}

//Agility
void Articuno::atk3() {
    cout << this->name << " used Agility " << endl;
    cout << this->name << " speed rose " << "\n" << endl;
    SPD *= 1.2;
}

//Blizzard
void Articuno::atk4(Pokemon &other) {
    cout << this->name << " attacks " << other.name << " with Blizzard " << endl;
    cout << "It's a critical hit!" << "\n" << endl;
    if (other.HP <= 110) {
        other.HP = 0;
        if (EXP + 75 < 100)
            EXP += 75;
        else
            EXP = (EXP + 75 - 100);
    }
    else
        other.HP -= 110;
}

```

2.14. Zapdos.h

```

#ifndef ZAPDOS.H
#define ZAPDOS.H

#include "Flying.h"
#include "Electric.h"

using namespace std;

//no dice que esta clase hereda de Electric y de Flying
class Zapdos : public Electric, public Flying {
public:
    string type;
    string strongVs;
    string weakVs;

public:

```

```

    Zapdos();
    Zapdos(string nombre);
    virtual ~Zapdos();

    void atk1 (Pokemon &other);
    void atk2 (Pokemon &other);
    void atk3 ();
    void atk4 (Pokemon &other);
    void print();

};

#endif // ZAPDOS.H

```

2.15. Zapdos.cpp

```

#include "Zapdos.h"

Zapdos::Zapdos() {

}

//implementacion del constructor del constructor
Zapdos::Zapdos(string nombre) {
    this->name = nombre;
    this->species = "Zapdos";
    this->type = Electric::type() + Flying::type();
    this->cry = "tnnrrnghhng";
    this->status = "Healthy";
    this->SPD = 100;
    this->HP = 90;
    this->ATK = 90;
    this->DEF = 85;
    this->sATK = 125;
    this->sDEF = 90;
    this->EXP = 0;
    this->strongVs = Electric::strongVs() + Flying::strongVs();
    this->weakVs = Electric::weakVs() + Flying::weakVs();
}

Zapdos::~~Zapdos() {

}

void Zapdos::print() {
    printInfo();
    cout << "Strong against: " << strongVs << endl;
    cout << "Weak against: " << weakVs << endl;
}

```

```

        cout << "\n";
    }

//Thunder Shock
void Zapdos::atk1(Pokemon &other) {
    cout << this->name << " attacks " << other.name << " with Thunder Shock " << endl;
    cout << "Enemy is paralyzed, Enemy's speed fell " << "\n" << endl;
    other.status = "Paralyzed";
    other.SPD *= 0.7;
}

//thunderbolt
void Zapdos::atk2(Pokemon &other) {
    cout << this->name << " attacks" << other.name << " with Flamethrower " << "\n" << endl;
    if (other.HP <= 90) {
        other.HP = 0;
        if (EXP + 75 < 100)
            EXP += 75;
        else
            EXP = (EXP + 75 - 100);
    }
    else
        other.HP -= 90;
}

//Agility
void Zapdos::atk3() {
    cout << this->name << " used Agility " << endl;
    cout << this->name << " speed rose " << "\n" << endl;
    SPD *= 1.2;
}

//Thunder
void Zapdos::atk4(Pokemon &other) {
    cout << this->name << " attacks " << other.name << " with Fire Blast " << endl;
    cout << "It's a critical hit!" << "\n" << endl;
    if (other.HP <= 110) {
        other.HP = 0;
        if ((EXP + 75) < 100)
            EXP += 75;
        else
            EXP = (EXP + 75 - 100);
    }
    else
        other.HP -= 110;
}

```

2.16. Moltres.h

```

#ifndef MOLTRESH
#define MOLTRESH

#include "Fire.h"
#include "Flying.h"

using namespace std;

//no dice que esta clase hereda de Fire y de Flying
class Moltres : public Fire, public Flying {
public:
    string type;
    string strongVs;
    string weakVs;

public:
    Moltres();
    Moltres(string nombre);
    virtual ~Moltres();

    void atk1 (Pokemon &other);
    void atk2 (Pokemon &other);
    void atk3 ();
    void atk4 (Pokemon &other);
    void print();

};

#endif // MOLTRESH

```

2.17. Moltress.cpp

```

#include "Moltres.h"

Moltres::Moltres() {

}

//implementacion del constructor del constructor
Moltres::Moltres(string nombre) {
    this->name = nombre;
    this->species = "Moltres";
    this->type = Fire::type() + Flying::type();
    this->cry = "ktnnnnrghhnnnnghhnnngn";
    this->status = "Healthy";
    this->SPD = 90;
    this->HP = 90;
    this->ATK = 90;
}

```

```

    this->DEF = 90;
    this->sATK = 125;
    this->sDEF = 85;
    this->EXP = 0;
    this->strongVs = Fire::strongVs() + Flying::strongVs();
    this->weakVs = Fire::weakVs() + Flying::weakVs();
}

Moltres::~~Moltres() {

}

void Moltres::print() {
    printInfo();
    cout << "Strong against: " << strongVs << endl;
    cout << "Weak against: " << weakVs << endl;
    cout << "\n";
}

//Burn up
void Moltres::atk1(Pokemon &other) {
    cout << this->name << " attacks " << other.name << " with Burn up " << endl;
    cout << "Enemy is burned " << "\n" << endl;
    other.status = "Burned";
    other.HP *= 0.85;
}

//Flamethrower
void Moltres::atk2(Pokemon &other) {
    cout << this->name << " attacks " << other.name << " with Flamethrower " << "\n";
    if (other.HP <= 90) {
        other.HP = 0;
        if (EXP + 75 < 100)
            EXP += 75;
        else
            EXP = (EXP + 75 - 100);
    }
    else
        other.HP -= 90;
}

//Agility
void Moltres::atk3() {
    cout << this->name << " used Agility " << endl;
    cout << this->name << " speed rose " << "\n" << endl;
    SPD *= 1.2;
}

```



```

//Fire Blast
void Moltres::atk4(Pokemon &other) {
    cout << this->name << " attacks " << other.name << " with Fire Blast " << endl;
    cout << "It's a critical hit!" << "\n" << endl;
    if (other.HP <= 110) {
        other.HP = 0;
        if (EXP + 75 < 100)
            EXP += 75;
        else
            EXP = (EXP + 75 - 100);
    }
    else
        other.HP -= 110;
}

```

3. Conclusiones

A modo de conclusión, se logró implementar en el lenguaje C++ un programa que crea objetos Pokemon así como ciertos pokemones en específico por medio de Herencia y Polimorfismo. Se creó una clase Pokemon luego se crearon 4 clases (Flying, Electric, Ice, Fire) que heredan de esta y luego 3 clases (Articuno, Zapdos, Moltres) que heredan de las 4 pasadas y por ende heredan de Pokemon. A cada pokemon se le crearon 4 ataques que tienen ciertos efectos. En el main se creó un pokemon de cada tipo y se atacaron unos a otros. A continuación se muestra la salida de la ejecución del programa.

```
Estado de los 3 pokemones creados

Nombre: Yigüirro
Especie: Moltres
Estado: Healthy
Speed: 90
Hp: 90
Special attack: 125
Special defense: 85
Experience: 0
Strong against: Grass Ice Bug Steel Grass Fight Bug
Weak against: Water Ground Rock Electric Ice Rock

Nombre: Pingüino
Especie: Articuno
Estado: Healthy
Speed: 85
Hp: 90
Special attack: 95
Special defense: 125
Experience: 0
Strong against: Grass Ground Flying Dragon Grass Fight Bug
Weak against: Fire Fight Rock Steel Electric Ice Rock

Nombre: Piolín
Especie: Zapdos
Estado: Healthy
Speed: 100
Hp: 90
Special attack: 125
Special defense: 90
Experience: 0
Strong against: Water Flying Grass Fight Bug
Weak against: Ground Electric Ice Rock
```

Figura 1: Se muestra el estado de los e Pokemones creados.

```
Fase de ataques

Pingüino attacks Piolín with Freezze dry
Enemy got frozen!

Nombre: Piolín
Especie: Zapdos
Estado: Frozen
Speed: 0
Hp: 90
Special attack: 125
Special defense: 90
Experience: 0
Strong against: Water Flying Grass Fight Bug
Weak against: Ground Electric Ice Rock
```

Figura 2: Se inicia la fase de ataques y se hace 1 ataque.

```
Pingüino used Agility
Pingüino speed rose

Nombre: Pingüino
Especie: Articuno
Estado: Healthy
Speed: 102
Hp: 90
Special attack: 95
Special defense: 125
Experience: 0
Strong against: Grass Ground Flying Dragon Grass Fight Bug
Weak against: Fire Fight Rock Steel Electric Ice Rock
```

Figura 3: Se hace otro ataque.

```
Yigüirro attacks Pingüino with Fire Blast
It's a critical hit!

Nombre: Pingüino
Especie: Articuno
Estado: Healthy
Speed: 102
Hp: 0
Special attack: 95
Special defense: 125
Experience: 0
Strong against: Grass Ground Flying Dragon Grass Fight Bug
Weak against: Fire Fight Rock Steel Electric Ice Rock

Nombre: Yigüirro
Especie: Moltres
Estado: Healthy
Speed: 90
Hp: 90
Special attack: 125
Special defense: 85
Experience: 75
Strong against: Grass Ice Bug Steel Grass Fight Bug
Weak against: Water Ground Rock Electric Ice Rock
```

Figura 4: Se hace otro ataque..