

פרויקט גמר בקורס תכנות מונחה עצמים בשפת C++

מגישים:

גל ארז – 313309213

ריף רוזנבלט – 206134645

גולן פרשה – 305001885

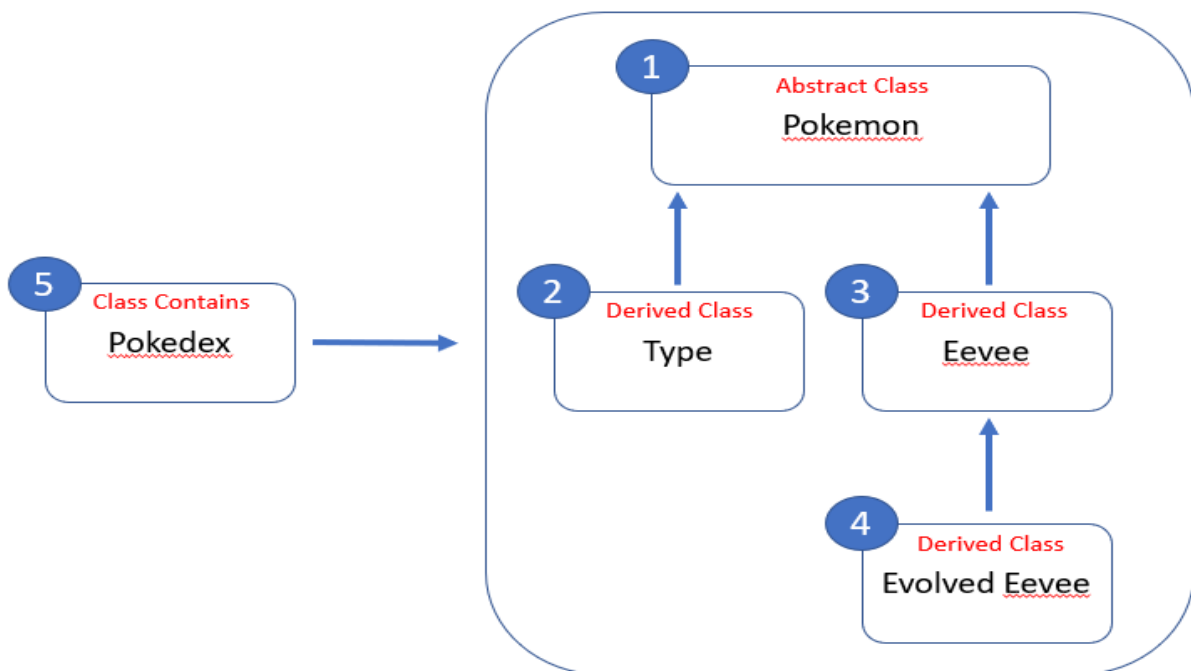


הסבר כללי:

הפרויקט עוסק בניהול נתונים תוך ביצוע פעולות אינטראקטיביות מול המשתמש, בנושא של עולם הפוקימונים.

Pokedex היא מערכת השומרת את המידע על הפוקימונים הנמצאים בידי המשתמש (ה-Trainer). בעולם הפוקימונים לכל מאמן צריך להיות מכשיר Pokedex על מנת להיות במעקב ולבצע פעולות בפוקימונים שלו, ולכן בחרנו לכתוב את הקוד ל-Pokedex.

מבנה הפרויקט:



מחלקת Pokemon

מחלקת הבסיס אשר מחזיקה בשם הפוקימון (PokemonName), כמות החיים שלו (Health Points – HP), רמתו (Level) ובוקטור המחזיק אפשרויות שונות לפיתוח הפוקימון בהמשך. בחרנו להשתמש דווקא בוקטור מכיוון שאין חשיבות לסדר המשתנים שבתוכו ולכן אין לנו צורך ברשימה.

בנאים ומחסל:

- בנאי ברירת מחדל המגדיר את השדות: PokemonName=Pikachu, HP=10, Level=1.
- בנאי המקבל ערכים מהמשתמש ובונה את הפוקימון.
- מחסל וירטואלי.

המתודות הקיימות במחלקה:

- מתודות Get: GetPokemonName, GetHP, GetLevel.
- מתודות Set: SetPokemonName (המשתמשת במתודה ValidateString על מנת לוודא שבערך שהוכנס אין מספרים), SetHP ו-SetLevel (מוודאות שלא הוכנס ערך שקטן או שווה לאפס).
- ValidateString – מקבלת מחרוזת ומוודאת שלא קיימים במחרוזת מספרים.
- PokeInfo – מתודה וירטואלית טהורה שמדפיסה את פרטי הפוקימון.

אופרטורים:

- אופרטור == שמקבל מצביע לפוקימון, מחזיר true אם הפוקימונים שווים או false אם הם אינם שווים.
- אופרטור << - אופרטור הדפסה של פרטי הפוקימון.

Pokemon.h

```
#pragma once
#include <iostream>
#include <string>
#include <vector>
using namespace std;

class Pokemon
{
protected:
    string PokemonName;
    int HP;
    int Level;
    vector<string>PossibleTypes = {"Fire", "Water", "Ground", "Electric", "Ice"};

public:
    Pokemon();
    Pokemon(const string PokemonName, int HP, int Level);
    virtual ~Pokemon();
    string GetPokemonName() const { return PokemonName; };
    int GetHP() { return HP; };
    int GetLevel() { return Level; };
    void SetPokemonName(const string Name)throw(exception);
    void SetHP(int HP)throw(exception);
    void SetLevel(int Level)throw(exception);
    virtual void validateString(const string s);
    virtual void PokeInfo(ostream& out) = 0;
    friend ostream& operator<<(ostream& out, const Pokemon& P);
    virtual bool operator==(const Pokemon* p);
};
```

Pokemon.cpp

```
#include <iostream>
#include <string>
#include "Pokemon.h"
using namespace std;

//Default constructor
Pokemon::Pokemon() : Level(1), HP(10), PokemonName("Pikachu")
{
}

//Constructor
Pokemon::Pokemon(const string PokemonName, int HP, int Level)
{
    this->SetPokemonName(PokemonName);
    this->SetHP(HP);
    this->SetLevel(Level);
}

//Destructor
Pokemon::~~Pokemon()
{
}

//Checks if the entered string doesn't have number in it
void Pokemon::validateString(const string s)
{
    if (s.empty())
    {
        throw exception("Name can't be NULL");
        exit(0);
    }
    else {
        for (int i = 0; i < s.length(); i++)
        {
            if (s[i] >= '0' && s[i] <= '9')
            {
                throw exception("Name can't have a number in it");
                return;
            }
        }
        return;
    }
}

//Checks if the name entered is valid and sets it as the Pokemon's name
void Pokemon::SetPokemonName(const string Name) throw(exception)
{
    validateString(Name);
    this->PokemonName = Name;
}

//Checks if the number entered is not 0 or lower and sets it as the Pokemon's HP
void Pokemon::SetHP(int HP) throw(exception)
```

```

{
    if (HP <= 0) throw exception("Your Pokemon is fainted");
    else
        this->HP = HP;
}

//Checks if the number entered is not 0 or lower and sets it as the Pokemon's level
void Pokemon::SetLevel(int Level) throw(exception)
{
    if (Level <= 0) throw exception("Level can't be lower than 1");
    else
        this->Level = Level;
}

//Checks if the Pokemons are equal
bool Pokemon::operator==(const Pokemon* p)
{
    if ((this->Level == p->Level) && (this->PokemonName == p->PokemonName) && (this->HP == p->HP))
        return true;
    else
        return false;
}

//Prints the information of the Pokemon
ostream& operator<<(ostream& out, const Pokemon& P)
{
    out << endl << "Pokemon's name: " << P.PokemonName << endl << "HP: " << P.HP <<
endl << "Level: " << P.Level;
    return out;
}

```

מחלקת Type

פוקימון מסוג ספציפי שהשתמש בוחר. זוהי מחלקה הנגזרת מ-Pokemon, המכילה את סוג הפוקימון (TypeName) שנבחר מהוקטור שבמחלקת Pokemon, מבנה של מתקפה מיוחדת (Special Attack), וחולשה (Weakness) שהיא מצביע לפוקימון מסוים מסוג Type הנמצא ב-Pokedex, אותו המשתמש יוכל להזין דרך התפריט בהמשך.

בנאים ומחסל:

- בנאי ברירת מחדל המגדיר את השדות: TypeName=Electric, SpecialAttack=1,n/a, Weakness=NULL.
- בנאי המקבל ערכים מהמשתמש ובונה את הפוקימון.
- בנאי העתקה המקבל פוקימון מסוג Type ומעתיק את ערכיו.
- מחסל וירטואלי.

המתודות הקיימות במחלקה:

- מתודות Get: GetTypeName, GetWeakness, GetSpecialAttack.
- מתודות Set: SetTypeName (המשתמש בוחר את ה-Type הספציפי מהוקטור שבמחלקת הבסיס), SetSpecialAttack.
- SetWeakness – מתודה המקבלת מצביע לפוקימון מסוג Type ומגדירה אותו כחולשה של הפוקימון.
- ReplaceSpecialAttack – מתודה המחליפה את המתקפה המיוחדת הקיימת.
- WhoWillWin – מתודה המקבלת מצביע לפוקימון מסוג Type, ובודקת למי יש Damage גבוה יותר. למי שיש את ה-Damage הגבוה יותר הוא זה שינצח בקרב.
- PokeInfo – מתודה המדפיסה את פרטי הפוקימון.

אופרטורים:

- אופרטור == שמקבל מצביע לפוקימון מסוג Type, מחזיר true אם הפוקימונים שווים או false אם הם אינם שווים.

Type.h

```
#pragma once
#include <iostream>
#include <vector>
#include <string>
#include "Pokemon.h"
using namespace std;

typedef struct SpecialAttack
{
    string AttackName;
    int Damage;
};

class Type : public Pokemon
{
private:
    string TypeName;
    SpecialAttack S_attack;
    Type* Weakness;

public:
    Type();
    Type(int Damage, const string Sname, const string Pokename, int HP, int Level);
    Type(const Type& t);
    ~Type() { delete Weakness; };
    string getTypeName() { return TypeName; }
    Pokemon* getWeakness() { return Weakness; }
    SpecialAttack getSpecialAttack() { return S_attack; }
    void setTypeNames()throw(exception);
    void setSpecialAttack(int Damage, const string Sname);
    void setWeakness(Type* pokemons);
    void ReplaceSpecialAttack();
    void WhoWillWin(Type* pokemon);
    void Pokemon::PokeInfo(ostream& out);
    bool operator==(const Type* p);
};
```


Type.cpp

```
#include "Type.h"

//Default constructor
Type::Type() :TypeName("Electric")
{
    this->setSpecialAttack(1, "n/a");
    this->Weakness = NULL;
}

//Constructor
Type::Type(int Damage, const string Sname, const string PokemonName, int HP, int Level)
{
    this->SetPokemonName(PokemonName);
    this->SetHP(HP);
    this->SetLevel(Level);
    this->setSpecialAttack(Damage, Sname);
    this->setTypeName();
    this->Weakness = NULL;
}

//Copy constructor
Type::Type(const Type& t)
{
    this->SetPokemonName(t.PokemonName);
    this->SetHP(t.HP);
    this->SetLevel(t.Level);
    this->setSpecialAttack(t.S_attack.Damage, t.S_attack.AttackName);
    this->TypeName = t.TypeName;
    this->Weakness = t.Weakness;
}

//Get's the possible types for the Pokemon and let's the user pick a specific type
void Type::setTypeName()
{
    int idx = 0;
    int choice;
    cout << endl << "Press the number of your wanted Pokemon's type: " << endl;
    while (idx < this->PossibleTypes.size())
    {
        cout << idx + 1 << " ) " << this->PossibleTypes[idx] << endl;
        idx++;
    }
    cin >> choice;
    if (choice<1 || choice>this->PossibleTypes.size())throw exception("Your choice
must be a valid number");
    this->TypeName = PossibleTypes[choice - 1];
}

//Sets the Pokemon's special attack
void Type::setSpecialAttack(int Damage, const string Sname)
{

```

```

        this->S_attack.Damage = Damage;
        this->S_attack.AttackName = Sname;
    }

    //Gets another Pokemon and sets it as the used Pokemon's weakness
    void Type::setWeakness(Type* weakness)
    {
        this->Weakness = weakness;
    }

    //Replaces the current special attack
    void Type::ReplaceSpecialAttack()
    {
        int newdmg;
        string newName;
        cout << endl << "Enter new special attack name: ";
        CLR_BFR
        getline(cin, newName);
        cout << "Enter new damage value: ";
        cin >> newdmg;

        setSpecialAttack(newdmg, newName);
        cout << endl << "special attack has been replaced" << endl;
    }

    //Gets another Pokemon and checks who will win in a battle by having a higher attack's
    damage
    void Type::WhoWillWin(Type* pokemon)
    {
        if (this->S_attack.Damage < pokemon->S_attack.Damage)
            cout << this->PokemonName << " won't beat " << pokemon->PokemonName <<
endl;
        if (this->S_attack.Damage == pokemon->S_attack.Damage)
            cout << "It's a tie" << endl;
        if (this->S_attack.Damage > pokemon->S_attack.Damage)
            cout << this->PokemonName << " will beat " << pokemon->PokemonName << endl;
    }

    //Checks if the Pokemons are equal
    bool Type::operator==(const Type* p)
    {
        if (this == p)
            return true;
        else
            return false;
    }

    //Prints the information of the Pokemon
    void Type::PokeInfo(ostream& out)
    {
        out << *this << endl << "Pokemon's type: " << this->TypeName << endl << "Special
attack and damage: " << this->S_attack.AttackName << ", " << this->S_attack.Damage <<
endl;
    }

```

מחלקת Eevee

פוקימון מסוג Eevee. זוהי מחלקה הנגזרת מ-Pokemon המכילה את הכינוי של הפוקימון (Nickname) ויכולת מיוחדת (Special Trait).

בנאים ומחסל:

- בנאי ברירת מחדל המגדיר את השדות: Nickname=Cutie, Special Trait=Flying, PokemonName=Eevee, HP=10, Level=1.
- בנאי המקבל ערכים מהמשתמש ובונה את הפוקימון.
- בנאי העתקה המקבל פוקימון מסוג Eevee ומעתיק את ערכיו.
- מחסל וירטואלי.

המתודות הקיימות במחלקה:

- מתודות Get: GetNickName, GetSpecialTrait.
- מתודות Set: SetNickName, SetSpecialTrait. שתי המתודות עושות שימוש ב-ValidateString על מנת לבדוק שלא הוכנסו מספרים למחרוזות.
- ChangeNickName – מתודה המשנה את הכינוי של הפוקימון.
- TrainEevee – מתודה המאמנת את הפוקימון ומעלה את הרמה שלו.
- PokeInfo – מתודה המדפיסה את פרטי הפוקימון.

אופרטורים:

- אופרטור == שמקבל מצביע לפוקימון מסוג Eevee, מחזיר true אם הפוקימונים שווים או false אם הם אינם שווים.

Eevee.h

```
#pragma once
#include <iostream>
#include <string>
#include <vector>
#include "Pokemon.h"
using namespace std;

class Eevee :public Pokemon
{
protected:
    string NickName;
    string SpecialTrait;

public:
    Eevee();
    Eevee(const string NickName, const string SpecialTrait, const string PokeName, int
HP, int Level);
    Eevee(const Eevee& e);
    ~Eevee();
    string getNickName() { return NickName; }
    string getSpecialTrait() { return SpecialTrait; }
    void setNickName(const string NickName) throw(exception);
    void setSpecialTrait(const string SpecialTrait) throw(exception);
    void ChangeNickName();
    void TrainEevee() throw(exception);
    void Pokemon::PokeInfo(ostream& out);
    bool operator==(const Eevee* p);
};
```

Eevee.cpp

```
#include "Eevee.h"

//Default constructor
Eevee::Eevee() :NickName("Cutie"), SpecialTrait("Flying"), Pokemon("Eevee", 10, 1)
{
}

//Constructor
Eevee::Eevee(const string NickName, const string SpecialTrait, const string PokemonName,
int HP, int Level)
{
    this->SetPokemonName(PokemonName);
    this->SetHP(HP);
    this->SetLevel(Level);
    this->setNickName(NickName);
    this->setSpecialTrait(SpecialTrait);
}

//Copy constructor
Eevee::Eevee(const Eevee& e)
{
    this->SetPokemonName(e.PokemonName);
    this->SetHP(e.HP);
    this->SetLevel(e.Level);
    this->setNickName(e.NickName);
    this->setSpecialTrait(e.SpecialTrait);
}

//Distructor
Eevee::~Eevee()
{
}

//Checks if the nickname entered is valid and sets it as Eevee's nickname
void Eevee::setNickName(const string NickName) throw(exception)
{
    Pokemon::validateString(NickName);
    this->NickName = NickName;
}

//Checks if the string entered is valid and sets it as Eevee's special trait
void Eevee::setSpecialTrait(const string SpecialTrait) throw(exception)
{
    Pokemon::validateString(SpecialTrait);
    this->SpecialTrait = SpecialTrait;
}

//Changes Eevee's nickname
void Eevee::ChangeNickName()
{
}
```

```

        string tmp;
        cout<<endl << "Please enter the new nickname: ";
        CLR_BFR
            getline(cin, tmp);
        setNickName(tmp);
    }

    //Training Eevee and raises it's level
    void Eevee::TrainEevee() throw(exception)
    {
        int AddLevel;
        cout<<endl << "For how many levels would you like to train " << this->PokemonName
        << "? ";
        cin >> AddLevel;
        if (AddLevel < 0)
            throw exception("Level cannot be a negative numebr");
        this->Level += AddLevel;
        cout << "The new level is: " << this->Level << endl;
    }

    //Checks if the Pokemons are equal
    bool Eevee::operator==(const Eevee* p)
    {
        if (this == p)
            return true;
        else
            return false;
    }

    //Prints Eevee's information
    void Eevee::PokeInfo(ostream& out)
    {
        out << *this << endl << "Eevee's nickname: " << this->NickName << endl << "Special
        Trait: " << this->SpecialTrait << endl;
    }

```

מחלקת EvolveEevee

פוקימון מפותח מסוג מיוחד של Eevee. זוהי מחלקה הנגזרת מ-Eevee המכילה את הסוג המיוחד של הפוקימון (SpecialType) וחפץ שהפוקימון מחזיק (ItemToHold).

בנאים ומחסל:

- בנאי ברירת מחדל המגדיר את השדות: Nickname=Flary, Special Trait=Flamethrower, PokemonName=Flareon, HP=50, Level=70, SpecialType=Fire, ItemToHold=Coin.
- בנאי המקבל ערכים מהמשתמש ובונה את הפוקימון.
- מחסל וירטואלי.

המתודות הקיימות במחלקה:

- מתודות Get: GetSpecialType, GetItemToHold.
- מתודות Set: SetSpecialType (שזורקת חריגה אם הוכנס מספר לא תקין), SetItemToHold (עושה שימוש ב-ValidateString על מנת לבדוק שלא הוכנסו מספרים למחרוזת).
- PokeInfo – מתודה המדפיסה את פרטי הפוקימון.

EvolveEevee.h

```
#pragma once
#include <iostream>
#include <string>
#include "Eevee.h"
using namespace std;

class EvolveEevee :public Eevee
{
private:
    string SpecialType;
    string ItemToHold;

public:
    EvolveEevee();
    EvolveEevee(const string ItemToHold, const string Nickname, const string
Specialtrait, const string Pokename, int HP, int Level);
    ~EvolveEevee();
    string getSpecialType() { return SpecialType; };
    string getItemToHold() { return ItemToHold; };
    void setSpecialType() throw(exception);
    void setItemToHold(const string ItemToHold) throw(exception);
    void Pokemon::PokeInfo(ostream& out);
};
```


EvolveEevee.cpp

```
#include "EvolveEevee.h"

//Default constructor
EvolveEevee::EvolveEevee() :SpecialType("Fire"), ItemToHold("Coin"), Eevee("Flary",
"Flamethrower", "Flareon", 50, 70)
{
}

//Constructor
EvolveEevee::EvolveEevee(const string ItemToHold, const string Nickname, const string
SpecialTrait, const string PokemonName, int HP, int Level)
{
    this->SetPokemonName(PokemonName);
    this->SetHP(HP);
    this->SetLevel(Level);
    this->setNickName(Nickname);
    this->setSpecialTrait(SpecialTrait);
    this->setSpecialType();
    this->setItemToHold(ItemToHold);
}

//Distructor
EvolveEevee::~EvolveEevee()
{
}

//Get's the possible types for the Pokemon and let's the user pick a specific type
void EvolveEevee::setSpecialType() throw(exception)
{
    int idx = 0;
    int choice;
    cout << "Choose a special type for Eevee: " << endl;
    while (idx < this->PossibleTypes.size())
    {
        cout << idx + 1 << " ) " << this->PossibleTypes[idx] << endl;
        idx++;
    }
    cin >> choice;
    if (choice<1 || choice>this->PossibleTypes.size())throw exception("Your choice
must be a valid number");
    this->SpecialType = PossibleTypes[choice - 1];
}

//Checks if the string entered is valid and sets it as the
void EvolveEevee::setItemToHold(const string ItemToHold) throw(exception)
{
    validateString(ItemToHold);
    this->ItemToHold = ItemToHold;
}
```

```
//Prints the Pokemon's information
void EvolveEevee::PokeInfo(ostream& out)
{
    out << *this << endl << "Nickname: " << this->NickName << endl << "Special trait:
" << this->SpecialTrait << endl << "Special type: " << this->SpecialType << endl << this-
>PokemonName << " is holding " << this->ItemToHold << endl;
}
```

מחלקת Pokedex

מחלקה מאגדת השומרת את שם המאמן (המשתמש) וכלל הפוקימונים שלו. הפוקימונים נשמרים בתוך וקטור של מצביעים מסוג Pokemon. בחרנו להשתמש בוקטור כיוון שזהו מערך דינאמי שנותן לנו את האפשרות לשנות את גודל המערך בקלות. מכיוון שאין חשיבות לסדר האיברים במערך, בחרנו להשתמש דווקא בוקטור ולא ברשימה. במתודת החיפוש חיפשנו מצביע מסוג Pokemon, מכיוון שכל שאר המחלקות הנגזרות הן גם מסוג Pokemon.

בנאים ומחסל:

- בנאי ברירת מחדל המגדיר את השדות: Trainer=Unknown.
- בנאי המקבל ערכים מהמשתמש ובונה את ה-Pokedex.
- בנאי העתקה המקבל Pokedex ומעתיק את שם המאמן.
- מחסל.

המתודות הקיימות במחלקה:

- מתודות Get: GetTrainerName.
- מתודות Set: SetTrainerName.
- PokeSearch – מתודה המקבלת שם של פוקימון ומחזירה מצביע מסוג Pokemon במידה והוא קיים ב-Pokedex. אם הפוקימון אינו קיים יחזור NULL.

אופרטורים:

- אופרטור += - מקבל פוקימון, בודק מאיזה סוג הוא ומכניס אותו לוקטור.
- אופרטור -= - מקבל פוקימון ועובר על איברי הוקטור. אם הפוקימון נמצא בוקטור האופרטור מוחק אותו.
- אופרטור <- - מדפיס את ה-Pokedex.

Pokedex.h

```
#pragma once
#include <iostream>
#include <string>
#include <vector>
#include "Pokedex.h"
#include "Eevee.h"
#include "Type.h"
#include "EvolveEevee.h"
using namespace std;

class Pokedex
{
private:
    vector<Pokemon*>PokeArr;
    string Trainer;

public:
    Pokedex();
    Pokedex(const string T);
    Pokedex(const Pokedex& P);
    ~Pokedex();
    string getTrainerName() { return Trainer; };
    void setTrainerName(const string Trainer);
    void operator+=(Pokemon& p);
    void operator-=(Pokemon& p);
    Pokemon* PokeSearch(const string PokeName);
    friend ostream& operator<<(ostream& out, Pokedex& p);
};
```

Pokedex.cpp

```
#include "Pokedex.h"

//Default constructor
Pokedex::Pokedex() :Trainer("Unknown")
{
}

//Constructor
Pokedex::Pokedex(const string T)
{
    this->setTrainerName(T);
}

//Copy constructor
Pokedex::Pokedex(const Pokedex& P)
{
    this->setTrainerName(P.Trainer);
}

//Destructor
Pokedex::~~Pokedex()
{
}

//Gets a string and sets it as the trainer's(the user) name
void Pokedex::setTrainerName(const string Trainer)
{
    this->Trainer = Trainer;
}

//Adds a Pokemon to the vector
void Pokedex::operator+=(Pokemon& p)
{
    if (strcmp(typeid(p).name(), "class Type")==0)
    {
        Type* tmp = dynamic_cast<Type*>(&p);
        this->PokeArr.push_back(&p);
    }
    if (strcmp(typeid(p).name(), "class Eevee")==0)
    {
        Eevee* tmp = dynamic_cast<Eevee*>(&p);
        this->PokeArr.push_back(&p);
    }

    if (strcmp(typeid(p).name(), "class EvolveEevee")==0)
    {
        EvolveEevee* tmp = dynamic_cast<EvolveEevee*>(&p);
        this->PokeArr.push_back(&p);
    }
}

//Erase a Pokemon from the vector
void Pokedex::operator-=(Pokemon& p)
```

```

{
    vector<Pokemon*>::iterator index;
    for (index = PokeArr.begin(); index != PokeArr.end(); index++)
    {
        if (*index == &p)
        {
            PokeArr.erase(index);
            return;
        }
    }
}

//Gets a Pokemon's name and checks if it exists in the vector
Pokemon* Pokedex::PokeSearch(const string PokeName)
{
    vector<Pokemon*>::iterator index;
    for (index = PokeArr.begin(); index != PokeArr.end(); index++)
    {
        if ((*index)->GetPokemonName() == PokeName)
            return *index;
    }
    return NULL;
}

//Prints the Pokedex
ostream& operator<<(ostream& out, Pokedex& p)
{
    if (p.PokeArr.empty())
    {
        cout << "Your pokedex is empty";
        return out;
    }
    else
    {
        for (int i = 0; i < p.PokeArr.size(); i++)
            p.PokeArr[i]->PokeInfo(out);
    }
}

```

Main

בפתיחת התוכנית, המשתמש נדרש להכניס את שם המאמן (את שמו של המשתמש) ואת מספר הפוקימונים שהוא רוצה להוסיף ל-Pokedex. תיפתח אפשרות לבחור את סוג הפוקימון – Normal Type, Eevee Type, Evolved Eevee Type. לאחר שהמשתמש בוחר את הסוג, ניתנת אפשרות לבנות את הפוקימון או ליצור פוקימון דיפולטיבי. במידה והמשתמש בחר בדיפולטיבי, יבנה הפוקימון בעזרת הבנאי ברירת מחדל המתאים לסוג הפוקימון שנבחר. אם המשתמש בחר לבנות בעצמו את הפוקימון, ייפתחו אפשרויות להכנסת ערכים בהתאם לסוג הפוקימון שנבחר. בניית הפוקימונים והכנסתם ל-Pokedex נעשית בעזרת מתודת עזר הנקראת AddPokemonToArr.

```
-----Welcome to your Pokedex!-----
Please enter your name: Ash Ketchum
How many Pokemon's would you like to add to the pokedex? 6
-----
Which kind of Pokemon would you like to add?
1) Normal type
2) Eevee type
3) Evolved Eevee type
1
Press 1 to create your own Pokemon or any other number to create a default Pokemon 1
Enter the Pokemon's name: Charmander
Enter Pokemon's HP: 50
Enter Pokemon's level: 10
Enter Pokemon's special attack name: Tail Weap
Enter Pokemon's special attack damage: 30
Press the number of your wanted Pokemon's type:
1) Fire
2) Water
3) Ground
4) Electric
5) Ice
1
Charmander Has been added to your Pokedex
-----
Which kind of Pokemon would you like to add?
1) Normal type
2) Eevee type
3) Evolved Eevee type
1
Press 1 to create your own Pokemon or any other number to create a default Pokemon 2
Pikachu has been added to your Pokedex
-----
Which kind of Pokemon would you like to add?
1) Normal type
2) Eevee type
3) Evolved Eevee type
2
Press 1 to create your own Eevee or any other number to create a default Pokemon 1
Enter Eevee's name: PokeEevee
Enter Eevee's HP: 30
Enter Eevee's level: 4
Enter Eevee's special trait: Charming
Enter Eevee's nickname:Awesome
PokeEevee has been added to your Pokedex
-----
```

```

Which kind of Pokemon would you like to add?
1) Normal type
2) Eevee type
3) Evolved Eevee type
2

Press 1 to create your own Eevee or any other number to create a default Pokemon 2
Eevee has been added to your Pokedex

-----

Which kind of Pokemon would you like to add?
1) Normal type
2) Eevee type
3) Evolved Eevee type
3

Press 1 to create your own Pokemon or any other number to create a default Pokemon 1

Enter Pokemon's name: Vaporeon
Enter Pokemon's HP: 100
Enter Pokemon's level: 50
Enter Pokemon's special trait: Water Gun
Enter Eevee's nickname: Splash
Enter item that Vaporeon holds: Water Stone
Choose a special type for Eevee:
1) Fire
2) Water
3) Ground
4) Electric
5) Ice
2

Vaporeon has been added to your Pokedex

-----

Which kind of Pokemon would you like to add?
1) Normal type
2) Eevee type
3) Evolved Eevee type
3

Press 1 to create your own Pokemon or any other number to create a default Pokemon 3

Flareon has been added to your Pokedex

-----

```

לאחר מכן ייפתח התפריט הראשי ובו 5 אפשרויות:

- פעולות עבור פוקימון ספציפי – המשתמש ידרש להכניס את שם הפוקימון בו ירצה להשתמש. התוכנית תחפש את הפוקימון בעזרת מתודת החיפוש PokeSearch. במידה ולא יימצא הפוקימון תופיע הודעה מתאימה, אם אכן יימצא הפוקימון ייפתח תפריט אפשרויות בהתאם לסוג הפוקימון.

```

-----
Choose an option:
1) Actions for specific Pokemon
2) Add Pokemon to Pokedex
3) Release a Pokemon
4) Print Pokedex
5) Exit
1

Enter the Pokemon's name you would like to use: Charmander

-----

```


- הוספת פוקימון לפוקדקס – תתחיל פעולת הוספת פוקימונים לפוקדקס כמו שהייתה בתחילת התוכנית.
- שחרור פוקימון – מחיקת פוקימון מהפוקדקס. יתבצע בעזרת האופרטור -=.
- הדפסת הפוקדקס – הדפסת כל פרטי הפוקימונים הנמצאים בפוקדקס.
- יציאה – סגירת התוכנית.

בבחירת פעולות עבור פוקימון מסוג Type יפתח התפריט הבא:

- החלפת מתקפה מיוחדת.
- בדיקה מי ינצח בקרב – תתבצע רק אם הפוקימון השני שהוכנס הוא גם מסוג Type. אם הוכנס פוקימון מסוג אחר תצא הודעה מתאימה.
- הוספת חולשה - תתבצע רק אם הפוקימון השני שהוכנס הוא גם מסוג Type. אם הוכנס פוקימון מסוג אחר תצא הודעה מתאימה.
- חזרה לתפריט הראשי.

```

-----
Choose an option:
1) Actions for specific Pokemon
2) Add Pokemon to Pokedex
3) Release a Pokemon
4) Print Pokedex
5) Exit
1
-----

Enter the Pokemon's name you would like to use: Charmander

Please choose an option:
1) Replace special attack
2) Check who will in a battle
3) Add a weakness
0) Back to main menu
1

Enter new special attack name: Ember
Enter new damage value: 60

special attack has been replaced
-----

Enter the Pokemon's name you would like to use: Charmander

Please choose an option:
1) Replace special attack
2) Check who will in a battle
3) Add a weakness
0) Back to main menu
2

Choose a Pokemon to battle with: Pikachu
Charmander will beat Pikachu
-----

Enter the Pokemon's name you would like to use: Pikachu

Please choose an option:
1) Replace special attack
2) Check who will in a battle
3) Add a weakness
0) Back to main menu
3

Choose a Pokemon that Pikachu will be weak against: Charmander

Pikachu is now weak against Charmander
-----

```

בבחירת פעולות עבור פוקימון מסוג Eevee ייפתח התפריט הבא:

- החלפת הכינוי של הפוקימון.
- לאמן את הפוקימון – המשתמש יכניס את המספר הרמות שהוא רוצה לאמן את הפוקימון, והרמה של הפוקימון תתעדכן בהתאם.
- חזרה לתפריט הראשי.

```
Enter the Pokemon's name you would like to use: Eevee
-----

Please choose an option:
1) Change Eevee's nickname
2) Train Eevee
0) Back to main menu
1

Please enter the new nickname: Pretty
Nickname has been changed
-----

Enter the Pokemon's name you would like to use: PokeEevee
-----

Please choose an option:
1) Change Eevee's nickname
2) Train Eevee
0) Back to main menu
2

For how many levels would you like to train PokeEevee? 10
The new level is: 14
-----
```

בבחירת פעולות עבור פוקימון מסוג EvolveEevee ייפתח התפריט הבא:

- החלפת הסוג המיוחד של הפוקימון – ייפתחו חמשת הסוגים שנמצאים בוקטור שבמחלקת Pokemon, מתוכם המשתמש יצטרך לבחור סוג מיוחד אחד.
- חזרה לתפריט הראשי.

```
Enter the Pokemon's name you would like to use: Vaporeon
-----

Please choose an option:
1) Change special type
0) Back to main menu
1

Choose a special type for Eevee:
1) Fire
2) Water
3) Ground
4) Electric
5) Ice
5

Special type has been replaced
-----
```

שחרור פוקימון:

```
Choose an option:
1) Actions for specific Pokemon
2) Add Pokemon to Pokedex
3) Release a Pokemon
4) Print Pokedex
5) Exit
3

Which Pokemon would you like to release? Eevee
Eevee was released

-----
```

הוספת פוקימון חדש:

```
Choose an option:
1) Actions for specific Pokemon
2) Add Pokemon to Pokedex
3) Release a Pokemon
4) Print Pokedex
5) Exit
2

-----

Which kind of Pokemon would you like to add?
1) Normal type
2) Eevee type
3) Evolved Eevee type
1

Press 1 to create your own Pokemon or any other number to create a default Pokemon 1

Enter the Pokemon's name: Squirtle
Enter Pokemon's HP: 30
Enter Pokemon's level: 6
Enter Pokemon's special attack name: Splash
Enter Pokemon's special attack damage: 20

Press the number of your wanted Pokemon's type:
1) Fire
2) Water
3) Ground
4) Electric
5) Ice
2

Squirtle Has been added to your Pokedex

-----
```

```
Choose an option:
1) Actions for specific Pokemon
2) Add Pokemon to Pokedex
3) Release a Pokemon
4) Print Pokedex
5) Exit
4

Pokemon's name: Charmander
HP: 50
Level: 10
Pokemon's type: Fire
Special attack and damage: Ember, 60

Pokemon's name: Pikachu
HP: 10
Level: 1
Pokemon's type: Electric
Special attack and damage: n/a, 1

Pokemon's name: PokeEevee
HP: 30
Level: 14
Nickname: Awesome
Special Trait: Charming

Pokemon's name: Vaporeon
HP: 100
Level: 50
Nickname: Splash
Special trait: Water Gun
Special type: Ice
Vaporeon is holding Water Stone

Pokemon's name: Flareon
HP: 50
Level: 70
Nickname: Flary
Special trait: Flamethrower
Special type: Fire
Flareon is holding Coin

Pokemon's name: Squirtle
HP: 30
Level: 6
Pokemon's type: Water
Special attack and damage: Splash, 20
```

```

Choose an option:
1) Actions for specific Pokemon
2) Add Pokemon to Pokedex
3) Release a Pokemon
4) Print Pokedex
5) Exit
5

Ash Ketchum, Thank you for using Pokedex! Hope to see you again soon!

C:\Users\galer\source\C++\FinalProject\Debug\FinalProject.exe (process 9340) exited with code 0.
Press any key to close this window . . .

```

Main.cpp

```

#include <iostream>
#include <string>
#include <vector>
#include <exception>
#include "Eevee.h"
#include "EvolveEevee.h"
#include "Pokedex.h"
#include "Pokemon.h"
#include "Type.h"

#define CLR_BFR fseek(stdin, 0, SEEK_SET);
using namespace std;

void AddPokemonToArr(Type* type, Eevee* eevee, EvolveEevee* Evolve, Pokedex* p)
{
    int choice, Dmg, HP, Level;
    string name, nickname, specialAttackName, specialTrait, itemToHold;
    do
    {
        cout << endl << "-----" << endl;
        cout << endl << "Which kind of Pokemon would you like to add?" << endl;
        cout << "1) Normal type" << endl;
        cout << "2) Eevee type" << endl;
        cout << "3) Evolved Eevee type" << endl;
        cin >> choice;

        if (choice < 1 || choice > 3)
            cout << "Unvalid number, please try again" << endl;
    } while (choice < 1 && choice > 3);

    switch (choice)
    {

```

```

case 1:
    cout << endl << "Press 1 to create your own Pokemon or any other number to
create a default Pokemon ";
    cin >> choice;
    if (choice == 1)
    {

        cout << endl << "Enter the Pokemon's name: ";
        CLR_BFR
        getline(cin, name);
        cout << "Enter Pokemon's HP: ";
        cin >> HP;
        cout << "Enter Pokemon's level: ";
        cin >> Level;
        cout << "Enter Pokemon's special attack name: ";
        CLR_BFR
        getline(cin, specialAttackName);
        cout << "Enter Pokemon's special attack damage: ";
        cin >> Dmg;
        try
        {
            type = new Type(Dmg, specialAttackName, name, HP, Level);
            cout << endl << name << " Has been added to your Pokedex " <<
endl;

        }
        catch (exception exp)
        {
            cout << "ERROR PUTTING VALUES!" << endl;
            exit(0);
        }
    }
    if (choice != 1)
    {
        type = new Type();
        cout << endl << "Pikachu has been added to your Pokedex " << endl;
    }
    *p += *type;
    return;

```

```

case 2:
    cout << endl << "Press 1 to create your own Eevee or any other number to
create a default Pokemon ";
    cin >> choice;
    if (choice == 1)
    {

        cout << endl << "Enter Eevee's name: ";
        CLR_BFR
        getline(cin, name);
        cout << "Enter Eevee's HP: ";
        cin >> HP;
        cout << "Enter Eevee's level: ";
        cin >> Level;
        cout << "Enter Eevee's special trait: ";
        CLR_BFR
        getline(cin, specialTrait);
        cout << "Enter Eevee's nickname: ";
        getline(cin, nickname);
        try

```

```

        {
            eevee = new Eevee(nickname, specialTrait, name, HP, Level);
            cout << endl << name << " has been added to your Pokedex " <<
endl;
        }
        catch (exception exp)
        {
            cout << "ERROR PUTTING VALUES!" << endl;
            exit(0);
        }
    }
    if (choice != 1)
    {
        eevee = new Eevee();
        cout << endl << "Eevee has been added to your Pokedex " << endl;
    }
    *p += *eevee;
    return;
case 3:
    cout << endl << "Press 1 to create your own Pokemon or any other number to
create a default Pokemon ";
    cin >> choice;
    if (choice == 1)
    {
        cout << endl << "Enter Pokemon's name: ";
        CLR_BFR
        getline(cin, name);
        cout << "Enter Pokemon's HP: ";
        cin >> HP;
        cout << "Enter Pokemon's level: ";
        cin >> Level;
        cout << "Enter Pokemon's special trait: ";
        CLR_BFR
        getline(cin, specialTrait);
        cout << "Enter Eevee's nickname: ";
        getline(cin, nickname);
        cout << "Enter item that " << name << " holds: ";
        getline(cin, itemToHold);
        try
        {
            Evolve = new EvolveEevee(itemToHold, nickname, specialTrait,
name, HP, Level);
            cout << endl << name << " has been added to your Pokedex " <<
endl;
        }
        catch (exception exp)
        {
            cout << "ERROR PUTTING VALUES!" << endl;
            exit(0);
        }
    }
    if (choice != 1)
    {
        Evolve = new EvolveEevee();
        cout << endl << "Flareon has been added to your Pokedex " << endl;
    }
    *p += *Evolve;
    return;

```

```

    }
}

void main()
{
    Pokedex* pokedex = new Pokedex();
    int amount, choice = 0;
    string trainer, p_name, PokemonToRelease;
    Eevee* eevee = NULL;
    EvolveEevee* evolve = NULL;
    Type* type = NULL;
    cout << "-----Welcome to your Pokedex!-----" << endl;
    cout << endl << "Please enter your name: ";
    getline(cin, trainer);
    pokedex->setTrainerName(trainer);
    cout << endl << "How many Pokemon's would you like to add to the pokedex? ";
    cin >> amount;
    for (int i = 0; i < amount; i++)
        AddPokemonToArr(type, eevee, evolve, pokedex);

    do
    {
        cout << endl << "-----" << endl;
        cout << endl << "Choose an option: " << endl;
        cout << "1) Actions for specific Pokemon" << endl;
        cout << "2) Add Pokemon to Pokedex " << endl;
        cout << "3) Release a Pokemon " << endl;
        cout << "4) Print Pokedex " << endl;
        cout << "5) Exit " << endl;
        cin >> choice;

        switch (choice)
        {
            case 1:
                do
                {
                    cout << endl << "-----" << endl;
                    cout << endl << "Enter the Pokemon's name you would like to
use: ";

                    CLR_BFR
                    getline(cin, p_name);
                    if (pokedex->PokeSearch(p_name) == NULL)
                    {
                        cout << "Couldn't find Pokemon's name";
                        break;
                    }

                    if (strcmp(typeid(*(pokedex->PokeSearch(p_name))).name(),
"class Type") == 0)
                    {
                        Type* ChosenPokemon = dynamic_cast<Type*>(pokedex-
>PokeSearch(p_name));

                        cout << endl << "Please choose an option:" << endl;
                        cout << "1) Replace special attack" << endl;
                        cout << "2) Check who will in a battle" << endl;
                        cout << "3) Add a weakness" << endl;
                        cout << "0) Back to main menu" << endl;
                    }
                }
            }
        }
    }
}

```



```

        cin >> choice;

        switch (choice)
        {
        case 1:
            ChosenPokemon->ReplaceSpecialAttack();
            break;

        case 2:
            cout << endl << "Choose a Pokemon to battle
with: ";

            CLR_BFR
            getline(cin, p_name);
            if (pokedex->PokeSearch(p_name))
            {
                if (strcmp(typeid(pokedex-
>PokeSearch(p_name)).name(), "class Type")==0)
                    cout << "You must choose a normal
type of Pokemon to compete with" << endl;
                else
                {
                    Type* BattlePokemon =
                    if (ChosenPokemon ==
                        cout << "The Pokemon can't
                        else
                        ChosenPokemon-
                    }
                }
            }
            else
                cout << "Couldn't find the Pokemon" <<
endl;
            break;
        case 3:
            cout << endl << "Choose a Pokemon that " <<
ChosenPokemon->GetPokemonName() << " will be weak against: ";
            CLR_BFR
            getline(cin, p_name);
            if (pokedex->PokeSearch(p_name))
            {
                if (strcmp(typeid(*(pokedex-
>PokeSearch(p_name))).name(), "class Type"))
                {
                    cout << "You must choose a normal
type of Pokemon" << endl;
                    break;
                }
                else
                {
                    Type* WeakPokemon =
                    if (ChosenPokemon == WeakPokemon)
                    {

```

```

        cout << "The Pokemon can't
be weak to itself" << endl;
        break;
    }
    else
    {
        ChosenPokemon->
        cout << endl <<
        ChosenPokemon->GetPokemonName() << " is now weak against " << WeakPokemon->
        GetPokemonName() << endl;
        break;
    }
}
}
else
    cout << "Couldn't find the Pokemon" <<
endl;
    break;
case 0:
    break;
}
}
else if (strcmp(typeid(*(pokedex->PokeSearch(p_name))).name(),
"class Eevee") == 0)
{
    Eevee* ChosenPokemon = dynamic_cast<Eevee*>(pokedex->
    PokeSearch(p_name));

    cout << endl << "-----" << endl;
    cout << endl << "Please choose an option:" << endl;
    cout << "1) Change Eevee's nickname" << endl;
    cout << "2) Train Eevee" << endl;
    cout << "0) Back to main menu" << endl;
    cin >> choice;

    switch (choice)
    {
    case 1:
        ChosenPokemon->ChangeNickName();
        cout << "Nickname has been changed" << endl;
        break;
    case 2:
        try
        {
            ChosenPokemon->TrainEevee();
        }
        catch (exception exp)
        {
            cout << "ERROR IN TRAIN EEEVE";
            exit(0);
        }
        break;
    case 0:
        break;
    }
}
else if (strcmp(typeid(*(pokedex->PokeSearch(p_name))).name(),
"class EvolveEevee") == 0)

```

```

        {
            EvolveEevee* ChosenPokemon =
dynamic_cast<EvolveEevee*>(pokedex->PokeSearch(p_name));
            cout << endl << "-----" << endl;
            cout << endl << "Please choose an option:" << endl;
            cout << "1) Change special type" << endl;
            cout << "0) Back to main menu" << endl;
            cin >> choice;

            switch (choice)
            {
            case 1:
                try
                {
                    ChosenPokemon->setSpecialType();
                    cout << endl << "Special type has been
replaced" << endl;

                }
                catch (exception exp)
                {
                    cout << "ERROR IN SETTING SPECIAL TYPE";
                    exit(0);
                }
                break;
            case 0:
                break;
            }
        }
    } while (choice != 0);
    break;
case 2:
    AddPokemonToArr(type, eevee, evolve, pokedex);
    break;
case 3:
    cout << endl << "Which Pokemon would you like to release? ";
    cin >> PokemonToRelease;
    if (pokedex->PokeSearch(PokemonToRelease))
    {
        *pokedex -= *pokedex->PokeSearch(PokemonToRelease);
        cout << PokemonToRelease << " was released " << endl;
    }
    else
        cout << "No such Pokemon exists " << endl;
    break;

case 4:
    cout << *pokedex;
    break;
case 5:
    cout << endl << pokedex->getTrainerName() << ", Thank you for using
Pokedex! Hope to see you again soon! " << endl;
    delete pokedex;
    exit(0);
}

} while (choice != 5);
}

```

