オブジェクト指向

はじめに…サイトで勉強しましたが、”なんとなく”しか理解できず、実感もわきませんでした。クラスの使い方などは他の二人が十分まとめていると思うので、私は中核概念を自己解釈でまとめると共に、実際に手を動かしてみたソースのコードを載せます。（習うより慣れたほうがいいかな・・・と思ったので）

* オブジェクト指向の中核概念
* 実装の隠蔽（カプセル化）
* 継承
* 多態性

これらが重要

* 実装の隠喩

外（クラス利用側）から見た振る舞いと中身（実装側）をわけて考える。

アクセスレベル….クラスのメンバのアクセス権利を制限することで、不正な書き込みを未然に防ぐ

(どこからでもアクセス可能…public

クラスの内部からのみアクセス可能…private)

Privateな値をmain等に取得したい場合、メソッドを通して取得する。

また、クラスの実装を変更した際に、利用側(たとえばmain)のコードはなるべく修正しなくてもいいようにする。

* 継承

実数の中に整数、人間の中に学生があるように、包含関係のあるものを表現する方法。派生クラス(学生)のインスタンスが生成される際、 派生クラスのコンストラクタが呼び出される前に 基底クラス(人間)のコンストラクタが呼び出される。

* 多態性

同じ名前のメソッドを呼び出しで、異なる振る舞いをすること。

・・・正直ここがよくわからない・・・同じような出力をしたいが違う処理にしなければならない時に使う?

* クラスと構造体との違い

メンバー変数やメソッドの定義は構造体でもできる。

|  |  |  |
| --- | --- | --- |
|  | 構造体 | クラス |
| 型の分類 | 値型 | 参照型 |
| 継承 | 不可能 | 可能 |
| 多態性 | 使えない | 使える |

構造体はクラスより限定的な使い方になるので、使えるならクラスを使ったほうがいいと思われる

* 作ってみたのだが

実装の隠喩と継承を使って、二次関数の解と、x=3の時の一次+数の値を出力するプログラムをC#で作ってみた・・・・・のだが、きれいに書けなかった。

(そもそもよく考えると一次関数が二次関数を継承する、という考え自体間違っていたのだが・・・)ここをこうすればきれいになる、という点や、オブジェクト指向の解釈が間違っている所があればどんなことでも言ってください。

ソースコード

class itijikannsuu//一次関数クラス

{

protected double kazu;//数の項保存、アクセス制限（protectedは内部と派生クラスのみに権利を与える）

protected double itiji;//一次の項保存、アクセス制限

public itijikannsuu()//コンストラクター

{

}

public void Suuzi(double suuji)//数の項セット

{

this.kazu = suuji;

}

public void One(double one)//一次の項セット

{

this.itiji = one;

}

public double xthree()//x=3の時の値を求める

{

return itiji\*3+kazu;

}

}

class nijikannsuu : itijikannsuu//二次関数クラス、継承

{

private double niji;//二次の項保存、アクセス制限

public nijikannsuu() : base() //コンストラクター

{

}

public void Two(double two)//二次の項セット

{

this.niji = two;

}

public double kai2a()//解1を求める

{

double a = 0;

double root = Math.Sqrt(itiji \* itiji - niji \* kazu \* 4);

a=(-itiji+root)/(niji\*2);//解の公式

return a;

}

public double kai2b()//解2を求める

{

double a = 0;

double root = Math.Sqrt(itiji \* itiji - niji \* kazu \* 4);

a = (-itiji - root) / (niji \* 2);//解の公式

return a;

}

}

class Class1//メインクラス

{

static void Main()

{

double two = double.Parse(Console.ReadLine());//入力

double one = double.Parse(Console.ReadLine());

double suuji = double.Parse(Console.ReadLine());

if (two == 0)

{

itijikannsuu b = new itijikannsuu();//初期化

b.Suuzi(suuji);//数の項セット、実装の隠喩(Mainを変えずに仕様変更できるようにする)

b.One(one);//一次の項セット

double iti = b.xthree();//x=3の時の値読み込み

Console.Write("x = 3の時の{0}x+{1}は{2}\n", one, suuji, iti);//出力

}

else

{

nijikannsuu z = new nijikannsuu();//初期化

z.Suuzi(suuji);//数の項セット(継承)

z.One(one);//一次の項セット(継承)

z.Two(two);//二次の項セット

double iti = z.xthree();//x=3の時の値読み込み(継承)

Console.Write("x = 3の時の{0}x+{1}は{2}\n", one, suuji, iti);//出力

double x1 = z.kai2a();//解1読み込み

double x2 = z.kai2b();//解2読み込み

Console.Write("{0}x^2+{1}x+{2}=0の時のx = {3},{4}\n", two, one, suuji, x1, x2);//出力

}

System.Threading.Thread.Sleep(10000);//10秒止める

}

}