プログラムの構造

クラス

変数 int index
変数 string str
メソッド hoge()
メソッド moge()

クラス

変数
変数
メソッド
メソッド

Classとはなにか

- ***データ構造を作る仕組み***
- **クラスを使うと新しいデータ型を作ることができる。**
- ***「オブジェクトを作る設計書」***

メソッドとはなにか

- 「オブジェクト指向プログラミングにおいて、オブジェクトに対する操作を定義した手続き」

- 簡単に言えば「処理」

メソッドの構成

[修飾子] 戻り値のデータ型 メソッド名(引数){
Public static void Main(string args){

修飾子 → **Public static**
戻り値のデータ型 → **void**
メソッド名 → **Main**
引数 → **string args(変数名argsの文字列)**

修飾子一覧

その他修飾子

修飾子	対象	メモ
abstract	クラス、メソッド	不完全な実装という指定になる。 オーバーライドして本実装をする必要がある時に指定する。
const	フィールド、変数	変更不可となる。（定数となる）
readonly	フィールド	読み取り専用となる。
static	クラス、フィールド、メソッド	静的メンバーとなる。 （newしなくてもアクセス可となる。）

アクセス修飾子

修飾子	意味	内容
public	公開	どこからでもアクセス可能
private	非公開	同一クラス内のみアクセス可能
internal	内部	自分自身のアセンブリ内部のクラスからのみ見える
protected	保護	派生クラスからアクセス可能
protected internal	内部保護	派生クラス or 自分のアセンブリ内部のクラスからアクセス可能

戻り値とは

メソッドの処理終了時に返す値のこと

ex1)public int aaa(){ **　return 1; } 戻り値1(int型)**

ex2)public string aaa(){ **　return “1”; } 戻り値1(string型)**

ex3)public float aaa(){ **　return 1f; } 戻り値1(float型)**

ex4)public void aaa(){ **　return; } 戻り値 無し**

データ型

型の種類	内容
byte	8bit整数(符号付)
int	32bit整数(符号付)
short	16bit整数(符号付)
long	64bit整数(符号付)
byte	8bit整数(符号なし)
ushort	16bit整数(符号なし)
uint	32bit整数(符号なし)
ulong	64bit整数(符号なし)
float	32bit浮動小数点
double	64bit浮動小数点
char	16bit Unicode
string	文字列(UTF-16)
bool	論理値 (true / false)
decimal	128bit型(高精度で科学計算などで使われる)
var	コンパイル時に型を自動で判定
object	全ての変数の基底オブジェクト型

void　型無し

データ型は色々定義できる

配列型、リスト、
クラス、ストラクト、列挙型などなど

Ex1)public List<int> aaa()
Ex2)public List<int> aaa()
Ex3)public Status bbb()
Ex4)public Elements ccc()
（メソッドだけでなく変数でも可）
Ex5)Status status;

```
Public Class Status()
{
Int hp;
Int mp;
}
Public enum Elements
{
AAA,
BBB
}
```

インスタンス

- インスタンス化：クラス（設計図）からオブジェクトを作ること
  ex)キャラクタークラス