Mind a PHP és a Java nyelvben is léteznek metódusok. PHP nyelvben eredetileg function-nak nevezik ezeke, és így a function kulcsszóval kerülnek bevezetésre. Java nyelvben a metódusokat nem kell ilyen módon bevezetni. Ezután a metódus neve és végül zárójelpárban a metódus paraméterlistája következik.

function metodus() {

}

Fontos különbség, hogy a láthatósági módosítók ugyanúgy jelen vannak a metódusoknál is, és ugyanazt jelentik. Így a Java nyelvben általában minden esetben kiírjuk a módosítókat, általában public, néhány esetben private módosítót szokás használni. Ezzel szemben PHP nyelvben általában publikus metódusokat használunk, és alapértelmezetten, ha nincs módosító akkor publikus lesz, tehát nem szokás módosítót kiírni.

Szintén fontos különbség, hogy, mivel a Java nyelv szigoruan tipusos, a PHP pedig nem, így a metódusoknál is megjelenik ez. A Java nyelvben a metódus neve elé ki kell írni a visszatérési érték típusát, míg a PHP nyelvben ez nem kell. Ugyanígy a paraméterlistában is a java nyelvben a változók elé a típust ki kell írni, míg PHP-ban, csak a változó nevét kell megadni.

Példa Java nyelvű metódusra:

public String newMethod(int paramA, String paramB) {

return paramB + “ “ + paramA;

}

Ugyanez PHP nyelven:

function newMethod($paramA, $paramB) {

return paramB.” “.$paramA;

}

A fenti példákból is látható, hogy a két nyelvben azonos, hogy a visszatérési értéket a return kulcsszóval kell megadni. A Java nyelvben minden esetben ugyanolyan típusú változóval lehet visszatérni, amilyen típust a metódus neve előtt megadtunk. PHP-ben ilyen megkötés nincs. Például az alábbi kód hiba nélkül lefut:

function newMethod($paramA, $paramB) {

return $paramA + $paramB;

}

echo newMethod(5, 2);

echo newMethod(5, “alma”);

Azonban az eredmény nem lesz midnen esetben hibátlan. Bár maga a kód nem fog hibát jelezni, az eredmény hibás lesz, mivel az szám és szöveg közötti összeadás PHP nyelven sincs értelmezve, viszont a gyenge tipusosság miatt ezt is végrehajtja, de az eredmény egyszerűen a szám lesz, illetve 0, ha minden paraméter szöveg. Így a programnak, illetve a programozónak kell erre figyelnie.

A Java nyelvben már szinte a kezdetektől osztályokat írunk, az objektum orientált programozás elveinek megfelelően, ezzel szemben a PHP nyelvben a mindez cask a 7-es verziótól kezdődően jelent meg, de manapság már ott is osztályokba szervezhetjük a programot.

Az osztályokat minden esetben a class kulcsszóval definiáljuk, ebben egyforma a két nyelv. Mint az eddigiekben is azonban, itt is a láthatósági módosítók különbözőek lesznek, lévén ugyanazok vannak mint a változók és metódusok esetében és ugyanolyan láthatóságot biztosítanak. Így a Java nyelvben ezeket ki kell írni, míg PHP nyelvben nem.

Példa osztály Java nyelven:

public class Osztaly {

}

Ugyanez PHP nyelven:

class Osztaly {

}

Az osztályoknak általános esetben vannak adattagjai, metódusai, esetleg konstruktora, destruktora. A változókat a Java nyelvben a korábban említettek alapján kell definiálni, meg kell adni egy módosítót, ami konvenció szerint privte legtöbbször, majd a változó típusát, és nevét. Ha azonnal inicializálni is akarjuk értéket is adhatunk neki, de ezt általában a konstruktorban szokás.

PHP nyelvben az osztályok változóit a var kulcsszóval definiáljuk, majd a korábbakhoz hasonlóat $nev változónevet adunk nekik.

A konstruktor, egy kiemelt metódus az osztályban, az osztály példányosításakor fut le egyszer, legtöbbször hazsnált célja az osztály változóinak inicializálása. Java nyelvben ha nem adunk meg konstruktort, akkor egy üres törzsű, ugynevezett default konstruktor fut le, mely csak egy dolgot csinál meghívja az ősosztály konstruktorát. Ha mi maunk írunk konstruktort, akkor a konstruktor metódus neve meg kell egyezzen az osztály nevével, általában public módosítóval kell ellátni, különben az osztályt nem lehet példányosítani így, és visszatérési értéke nincs.

PHP nyelvben a konstruktor hasonló funkcióval bír, paraméterként kaphat értékeket, melyekkel inicializálni tudja az osztály változóit. PHP-ban nincs default konstruktor, így ha nem írunk konstruktort, nem fog lefutni semi.

Az osztályban írhatunk metódusokat, a fentebb említett módon. Ha public módosítóval látjuk el őket, akkor az osztályon kívülről meghívhatók, Java nyelvben az objectum.metodusNev() utasítással, míg PHP nyelvben az objektum->metudosNev() utasítással.

Példa egy osztályra Java nyelvben:

Public class Osztály {

private int paramOne;

private int paramTwo;

public Oszaly(int paramOne, int paramTwo) {

this.paramOne = paramOne;

this.paramTwo = paramTwo;

}

public int getParamOne() {

return this.paramOne;

}

public void setParamOne(int paramOne) {

this.paramOne = paramOne;

}

}

Ezután a futtatható osztályban pedig:

Osztaly o = new Osztaly(1, 5);

o.setParamOne(o.getParamOne()\*5);

Ugyanez az osztály PHP nyelven:

class Osztaly {

var $paramOne;

var $paramTwo;

function \_\_construct($paramOne, $paramTwo) {

$this->paramOne = $paramOne;

$this->paramTwo = $paramTwo;

}

function getParamOne() {

return $this->paramOne;

}

function setParamOne($paramOne) {

$this->paramOne = $paramOne;

}

}

Ezután példányosításkor:

$oszaly = Osztaly(5, 13);

$osztaly->setParamOne($osztaly->getParamOne()\*15);