

# PHP PROGRAMOZÁS

1. forduló



A kategória támogatója: Human Priority Kft.

## Ismertető a feladathoz

**Kérjük, hogy a feladatlap indítása előtt mindenképp olvasd el az alábbi útmutatót:**

- MINDEN kérdésre **van helyes válasz**.
- Olyan kérdés **NINCS**, amire az összes válasz helyes, ha mégis az összes választ bejelölöd, arra a feladatra automatikusan 0 pont jár.
- A **radio button-os** kérdésekre **egy helyes válasz van**.
- **Ha lejár a feladatlap ideje, a rendszer AUTOMATIKUSAN** beküldi azt az addig megjelölt válaszokkal.
- Azokat a feladatlapokat, amelyekhez **csatolmány** tartozik, javasoljuk **NEM mobilon** elindítani, erre az érintett feladatlapok előtt külön felhívjuk a figyelmet.
- Az **adatbekérős feladatokra NEM jár részpontszám**, csak a feleletválasztósakra.
- **Helyezéseket a 4. forduló után mutatunk**, százalékos formában: adott kategóriában a TOP 20-40-60%-hoz tartozol.
- **Badge-eket** szintén a 4.forduló után kapsz majd először.
- Ha egyszerre több böngészőből, több ablakban vagy több eszközről megnyitod ugyanazt a feladatlapot, **nem tudjuk vállalni** az adatmentéssel kapcsolatban esetlegesen felmerülő anomáliákért a felelősséget!
- A hét forduló során az egyes kategóriákban (de nem feltétlenül mindegyikben) **könnyű-közepes-nehez kérdésekkel** egyaránt találkozhatasz majd.

**Jó versenyzést kívánunk!**

Felhasznált idő: 15:31/25:00

Elért pontszám: 9/11

## 1. feladat 1/1 pont

Az alábbi kódrészletet futtatva mi fog megjelenni a kimeneten?



```
$var = 'false';

if ($var) {
    echo 'true';
} else {
    echo 'false';
}
```

### Válasz



true

Ez a válasz helyes, és meg is jelölted.



false

### Magyarázat

A **\$var** változó egy stringet kap értékül, így a feltétel kiértékelésekor az IGAZ ágba jutunk, hiszen a változó létezik és van értéke.

## 2. feladat 0/0 pont

Melyik üzenetet fogja megjeleníteni az alábbi hibakezelő osztály?

```

class MyCustomException extends Exception {

    try {
        throw new MyException('Error!');
    } catch (Exception $e) {
        echo 'Exception type: Exception';
    } catch (MyCustomException $e) {
        echo 'Exception type: MyCustomException';
    }

}

```

## Válasz

☒ Exception type: Exception

Ez a válasz helyes, és meg is jelölted.

☐ Exception type: MyCustomException

## Magyarázat

Kedves Versenyzők!

A kérdést 0 pontosra állítottuk a visszajelzéseitek alapján a feladatírókkal való egyeztetést követően.

A feladatban szereplő kódrészlet egyáltalán nem futtatható abban a formában, ahogyan le van írva. Az csak egy pseudo-kód, aminek célja, hogy az ún. *Exception*-ök lekezelésének a sorrendjére kérdezzen rá. A „*MyException*” osztály valóban nincs feltüntetve a kódban, ily módon ténylegesen futásidejű hiba keletkezhet a kódban, tehát egyik *Exception* üzenete sem fog megjelenni.

Részünkről a legkorrektebb az lett volna, ha a kódrészletben kifejtjük a *MyException* osztályt is, vagy a kérdés szövegezésében jelezhattük volna, hogy azzal a feltételezéssel élünk, hogy a *MyException* osztály létezik és az adott kódban elérhető.

Köszönjük megértéseket!

A példában két *catch* ág van és az *Exception* *catch* ága előbb szerepel, mint a *MyCustomException* *catch* ága. Ugyan *MyException* kivétel keletkezett a *MyCustomException* osztályban, ez az osztály azonban az *Exception* osztályból származik, így annak *catch* ága is érvényesül. Amint egy *catch* ág érvényesült, a további *catch* ágak már nem hajtódnak végre.

Amennyiben fordítva írnánk a sorrendet, azaz a *MyCustomException* *catch* ága szerepelne az első helyen, úgy a *MyCustomException* *catch* ága hajtódnak volna végre.

## 3. feladat 0/2 pont

Mi lesz a *\$result* változó értéke?



```
$consoles = ['PlayStation', 'Xbox', 'PC', 'Nintendo'];  
$result = next($consoles);  
$result = reset($consoles);  
$result = prev($consoles);  
$result = next($consoles);  
$result = next($consoles);  
$result = current($consoles);
```

### Válasz

☐ PlayStation

☒ Xbox

Ez a válasz helytelen, de megjelölted.

☐ PC

☐ Nintendo

☒ (bool) false

Ez a válasz helyes, de nem jelölted meg.

### Magyarázat

A **reset** parancs az első tömbelemre (PlayStation) állítja vissza a belső mutatót, így a következő **prev** parancs nem fogja tudni megtalálni az ezt megelőző elemet. Ekkor egy logikai **false** értéket ad vissza, a további tömbkezelő parancsok pedig már nem tudják átállítani a pointert, így a **\$result** változó értéke false marad.

## 4. feladat 4/4 pont

Adott az alábbi Car osztály:

```
class Car {
    protected $name = '';
    protected $horsePower = 100;

    public function getName() {
        return $this->name;
    }

    public function getHorsePower() {
        return $this->horsePower;
    }
}
```

A következő kódrészletet fogjuk meghívni:

```
$raceCar = new RaceCar();
echo $raceCar->getName() . ' has ' . $raceCar->getHorsePower() . ' hp.';
```

Azt szeretnék, hogy a kimenet a következő legyen: **"Ferrari has 300 hp."**

Válaszd ki a helyes megoldást vagy megoldásokat!

### Válaszok

☒ 1.válasz:

```
class RaceCar extends Car {  
  
    public function getName() {  
        return 'Ferrari';  
    }  
  
    public function getHorsePower() {  
        return 300;  
    }  
}
```

Ez a válasz helyes, és meg is jelölted.

☐ 2.válasz:

```
class RaceCar extends Car {  
    private $name = 'Ferrari';  
    private $horsePower = 300;  
}
```

☒ 3.válasz:

```
class RaceCar extends Car {  
    protected $name = 'Ferrari';  
    protected $horsePower = 300;  
}
```

Ez a válasz helyes, és meg is jelölted.

☐ 4.válasz:

```
class RaceCar {  
    protected $name = 'Ferrari';  
    protected $horsePower = 300;  
}
```

## Magyarázat

A helyes válaszok az 1. és a 3.

Az 1. esetén felülírtuk a protected osztályváltozókat, majd meghívtuk a getter metódusokat a Carős-osztályból, melyek kiírják a protected változók mindenkori értékét.

A 3. esetben magát a getter metódusokat írtuk felül, melyek így közvetlenül visszaadják a kívánt értékeket.

## 5. feladat 4/4 pont

Hogyan tudunk írásra megnyitni egy fájlt, úgy, hogy az új tartalmat a fájl meglévő tartalmához fűzzük hozzá?

### Válasz

☐

```
fopen('c:\\folder\\resource.txt', 'r');
```



```
fopen('c:\\folder\\resource.txt', 'w');
```



```
fopen('c:\\folder\\resource.txt', 'a');
```

Ez a válasz helyes, és meg is jelölted.



```
fopen('c:\\folder\\resource.txt', 'w+');
```

## Magyarázat

Az **"a"** paraméter írásra nyitja meg a fájlt, és egyúttal a fájlmutatót is a fájl végére helyezi. A **"w"** és **"w+"** paraméterek ugyan írásra nyitják meg a fájlt, de a fájlmutatót a fájl elejére teszik, így a tartalom felülíródna. Az **"r"** paraméter pedig csak olvasásra nyitja meg a fájlt.





