

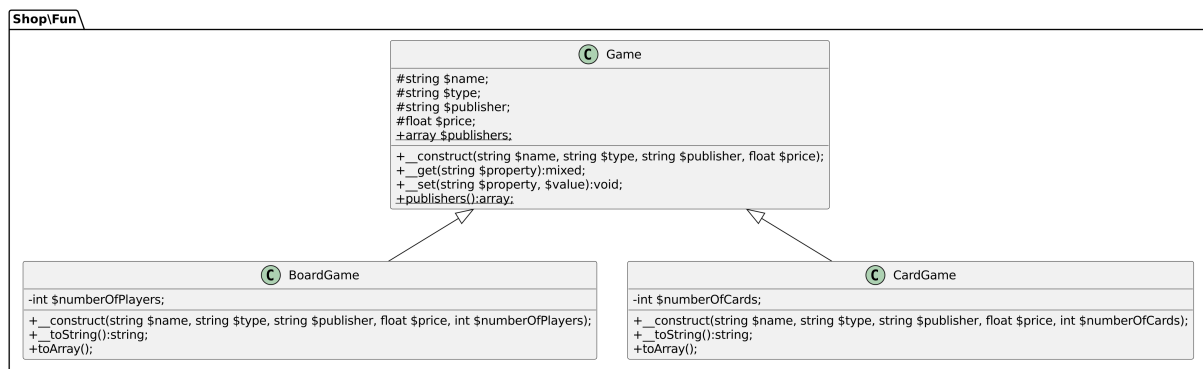
Games

OOP, öröklődés, parent

Rostagni Csaba

Előkészületek

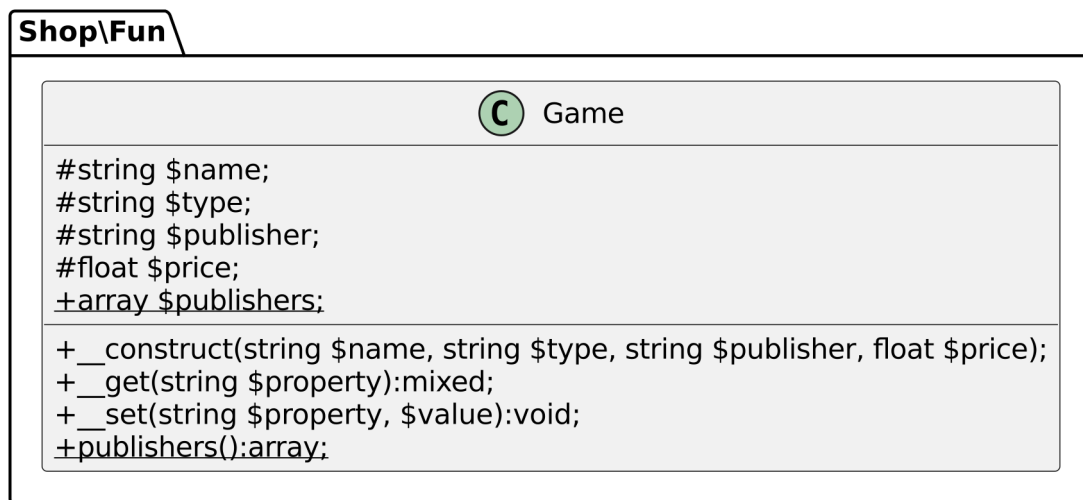
1. Hozzon létre egy projektmappát **games-vezeteknev-keresztnev** néven.
2. Hozzon létre a projektmappában egy fájlt **games.php** néven. Ez lesz a konzolos szkript belépési pontja.
 - Állítsa be, hogy a szkript szigorúan kezelje a típusokat!
 - Minden hibát figyelmeztetést, stb. . . jelenítsen meg!
3. Hozzon létre egy mappát **src** néven. A saját készítésű osztályok itt helyezkedjenek el a névtér szerint úgy, hogy a PSR-4-es szabványak feleljenek meg!
4. Ahogy az UML ábrán is látható, a *vendor névtér* **Shop** legyen. Hozza létre neki az **Shop** mappát, majd készítsen egy olyan **composer.json** fájlt, ahol az autoloader PSR-4 szabvány szerint a **Shop** névtérhez tartozó kódot az **src/Shop** mappában keresi.



1. ábra. Osztályok kapcsolata

Osztályok

Game



2. ábra. Game

5. Vegye fel a `name`, `type`, `publisher` és `price` tulajdonságokat az UML ábrán látható láthatósággal és típussal.
6. Bővítse az osztályt egy osztály szintű `publishers` tulajdonsággal, ami a *“Hasbro”*, *“Mattel”* és *“Asmodee”* értékekett tartalmazza.
7. Hozzon létre egy konstruktort, ami rendre a `name`, `type`, `publisher` és `price` paramétereket kapja meg, és állítja be az osztály megfelelő tulajdonságát. Jelölje a megfelelő típusokat!
8. Hozzon létre a `__get()` és `__set()` metódusokat a tanultak alapján. Ügyeljen a típusokra!

BoardGame

Shop\Fun

C BoardGame

```
-int $numberOfPlayers;
+__construct(string $name, string $type, string $publisher, float $price, int $numberOfPlayers);
+__toString():string;
+toArray();
```

3. ábra. Game

9. Származtassa a **BoardGame** osztályt a **Game** osztályból.
10. Hozzon létre egy konstruktort, ami rendre a **name**, **type**, **publisher**, **price** és **numberOfPlayers** paramétereket kapja meg. Ne felejtse el az örökölt tulajdonságokat a szülő konstruktorával inicializálni!
11. Valósítsa meg a **Stringable** interfészt. Bővítse az osztályt egy **__toString()** metódussal, ami az alábbi minta szerint adja vissza az objektum adatait, ahol a **<** és **>** jeleket a megfelelő tulajdonság jelenjen meg. *A sortörés csak az olvashatóság miatt került a leírásba, a kódban **ne** szerepeljen.*

```
<publisher> <name> <type>
board game (<numberOfPlayers> players) <price> EUR
```

például:

```
Asmodee The Board Game 3 strategy board game (4 players) 45.99 EUR
```

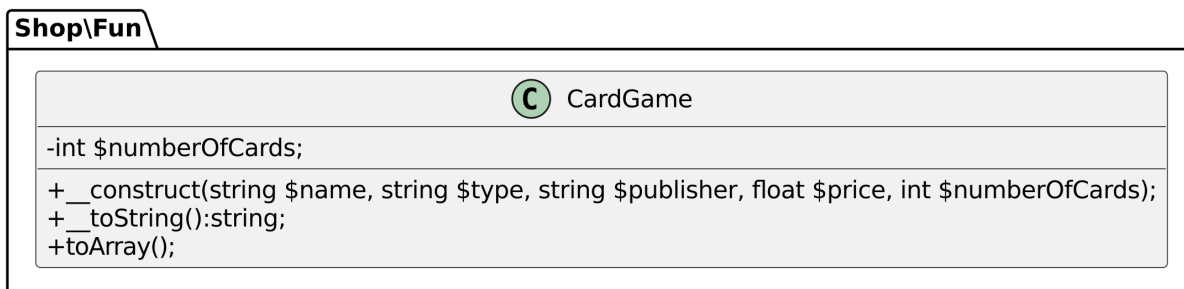
12. Bővítse az osztályt egy **toArray()** metódussal, ami az objektum tulajdonságait adja vissza tömbként az alábbi sorrendben:

```
[<publisher>, <name>, <type>, <numberOfPlayers>, <price>]
```

például:

```
["Asmodee", "The Board Game 3", "strategy", 4, 45.99]
```

CardGame



4. ábra. Game

13. Származtassa a **CardGame** osztályt a **Game** osztályból.
14. Vegye fel a **numberOfCards** tulajdonságot az UML ábra szerint.
15. Hozzon létre egy konstruktort, ami rendre a **name**, **type**, **publisher**, **price** és **numberOfCards** paramétereket kapja meg. Ne felejtse el az örökölt tulajdonságokat a szülő konstruktorával inicializálni!
16. Valósítsa meg a **Stringable** interfészt. Bővítse az osztályt egy **__toString()** metódussal, ami az alábbi minta szerint adja vissza az objektum adatait, ahol a **<** és **>** jeleket a megfelelő tulajdonság jelenjen meg. *A sortörés csak az olvashatóság miatt került a leírásba, a kódban **ne** szerepeljen.*

```

<publisher> <name> <type>
card game (<numberOfCards> cards) <price> USD
  
```

például:

```

Asmodee TheCard Game 1 deck building card game (52 cards) 25.99 EUR
  
```

17. Bővítse az osztályt egy **toArray()** metódussal, ami az objektum tulajdonságait adja vissza tömbként az alábbi sorrendben:

```

[<publisher>, <name>, <type>, <numberOfCards>, <price>]
  
```

például:

```

["Asmodee", "TheCard Game 1", "deck building", 52, 25.99]
  
```

Konzolos szkript

18. A `games.php` lesz a szkript belépési pontja.
19. Hozzon létre egy mappát `out` néven. Ez lesz a kimeneti fájlok helye.
20. A szkript futtatásához legalább 1 paraméter szükséges.
21. Amennyiben túl kevés paramétert ad meg a felhasználó, úgy figyelmeztesse a szkript, majd lépjen ki 4-es hibakóddal.
22. Az első paraméter csak `boardgame` vagy `cardgame` lehet. Ez határozza meg, hogy a szkript milyen típusú objektumokat generáljon. Amennyiben bármilyen más adatot kapna, úgy adjon hibaüzenetet, majd a 49-es hibakóddal lépjen ki!
23. Hozzon létre egy tömböt `$games` néven. Generáljon bele 6 társasjátékot, **vagy** 6 kártyajátékot a **paraméterezésének megfelelően!**
 - A kiadó a `Game` osztályban található egyik kiadó legyen.
 - A játék neve két szóból és egy 1 és 9 közötti számból áll.
 - A típus társasjátékok esetében *“strategy”* legyen.
 - A társasjátékok ára 20 és 100 EUR közötti érték legyen, a szélsőértékeket is beleértve.
 - A társasjátékok esetében a játékosok száma legyen 1 és 6 között.
 - A típus kártyajátékok esetében *“deck building”* legyen.
 - A kártyajátékok ára 10 és 50 EUR közötti érték legyen, a szélsőértékeket is beleértve.
 - A kártyajátékok esetében a kártyák száma legyen 52.
24. Amennyiben csak 1 paramétert kapott a szkript, úgy jelenítse meg az összes generált játékot a megfelelő `__toString` felhasználásával.
25. Amennyiben a második paraméter értéke `csv`, úgy az `out` mappába hozzon létre a típusnak megfelelő nevű `boardgame.csv` vagy `cardgame.csv` fájlt, ami a generált játékok adatait tartalmazza, **pontosvesszőkkel** elválasztva az egyes sorokban.