

Dizájn dokumentum

Rövid leírás

A program egy játék karakterének/karaktereinek teszi lehetővé, hogy egy tárgyat vásároljon a boltból, amely tárgy bónuszai növelik a karakter(ek) tulajdonságait.

A program szerkezete

- Karakter létrehozás: A felhasználó 2 karaktertípus közül választhat (Mágus, Harcos)
- Vásárlás a boltból: A felhasználónak felkínál a bolt 3 válaszlehetőséget, mindhárom tárgy különböző mértékű tulajdonságnövelést biztosít.
- Pálya létrehozás: A felhasználó 3 különböző méretű pályát hozhat létre

Használt tervezési minták

Factory Method

A Gyártómetódust használom a karakterek létrehozására. Azért jó ez a tervezési minta, mert itt elkészítés algoritmusát egységbe tudjuk zárni, így ha a gyártás folyamata változik, akkor azt csak egy helyen kell módosítani. Az ösosztályban elhelyezett gyártómetódus írja le a gyártás algoritmusát, a gyermek osztály pedig eldönti, hogy mit kell gyártani. Ennek érdekében az algoritmus háromféle lépést tartalmazhat:

- A gyártás közös lépései
- A gyártás kötelező változó lépései.
- A gyártás opcionális lépései

Karakter létrehozás esetén a programban:

A CharacterCreate osztályban írom le a gyártás algoritmusát.

```
8 references
abstract class CharacterCreate
{
    0 references
    public void create()
    {
        Choose();
        Create();
    }
    3 references
    protected abstract void Choose();
    3 references
    protected abstract void Create();
}
```

A programomban csak kötelező változó lépéseket használtam amelyek a következők:

- Choose()
- Create()

Decorator Pattern

A díszítő tervezési mintát arra használom, hogy a felhasználónak lehetősége legyen a boltban tárgyat vásárolni és a tárgy hatással legyen a karakter tulajdonságaira. Azért hasznos ez a tervezési minta, mert egy alap objektumot tudunk vele „feldíszíteni” és könnyen cserélhetők a „díszítőelemek”. Ebben a programban a díszítendő alap az alap karakter, amelynek tulajdonságai vannak előre megadott értékkel és ezt az alapot díszítem fel tárgyakkal, amik változtatják a tulajdonságok értékeit. Jelen esetben 3 díszítő elemet készítettem el, de a program kód minimális módosításával bármennyi díszítőt lehet készíteni. Ennél a szerkezeti tervezési mintánál az absztrakt ősből indulunk ki, amely a kódban a CharacterBase osztály. Az alap karaktert, díszíteni lehet különböző díszítő osztályokkal. Ezek a kódban tárgyak, amelyek véletlenszerűen generált értékekkel módosítják a karakter értékeit.

Strategy Pattern

A stratégia tervezési mintát azért használtam, mert van változékony metódusom, ami jelen esetben a MapGen(), mert a különböző méretű pályákat másképp generáljuk le. A stratégiai tervezési mintát használva a változékony metódust kiemeljük egy osztályhierarchiába. Szétválasztom az osztályt és a változékony metódust, majd a szétválasztott dolgokat újra összerakom az objektum-összetétel segítségével.

Rendszer működése

1. A felhasználónak felajánlom, hogy két karakter közül válasszon, ha helytelenül választ újra felajánlom.
2. A felhasználó tetszőleges számú karaktert választhat, aminek a költségét egy listában tárolom.
3. Ha a felhasználó már nem választ több karaktert, akkor a boltból vásárolhat egy tárgyat.
4. Felajánlom a felhasználónak, hogy a három különböző méretű pályából válasszon.
5. Létrehozom a pályát és kiíratom, hogy mennyibe került a karakter, illetve a tárgy vásárlása.

```
Please choose a character and write (a/b)!
Choosable characters:
a) Magician
b) Warrior
a
Want to buy another character? If yes write yes.
0
Please choose one item.
a) 1
b) 2
c) 3
c
Please choose a map.
a) small
b) medium
c) big
c
Big map created.

800 Point is the cost.
```