

Korszerű Programozási Technikák

NagyZH (pótlás) – 2017.12.19.

Név:

Neptun kód:

A feladat egy alakzatokat előállító gyárhoz beérkezett rendelések feldolgozása. A megrendelt alakzatokat előre létrehozott prototípusok alapján gyártják. A következő osztályok adottak:

- **Shape**: egy alakzatot reprezentáló absztrakt osztály a következő adattagokkal: *color* (alakzat színe, *string*), *size* (alakzat mérete, *string*), *type* (alakzat típusa, *string*, lehetséges értékek: **Circle**, **Triangle**, **Rectangle**). Az adattagok konstruktorral állíthatók be, getterekkel lekérhetők és setterekkel módosíthatók. A *setParameters* virtuális metódust a származtatott osztályok valósítják meg: a gyerekosztályra jellemző adattagokat állítja be a függvény,
- **Circle**: egy kör alakzatot reprezentáló osztály, a **Shape**-ből származik, plusz adattagja a kör sugara (*radius*, *int*), a *setParameters* ezt állítja be,
- **Rectangle**: egy téglalap alakzatot reprezentáló osztály, a **Shape**-ből származik, plusz adattagjai a szélesség (*width*, *int*) és magasság (*height*, *int*), a *setParameters* ezeket állítja be,
- **Triangle**: egy háromszög alakzatot reprezentáló osztály, a **Shape**-ből származik, plusz adattagjai az alap (*base*, *int*) és magasság (*height*, *int*), a *setParameters* ezeket állítja be,
- **Order**: egy rendelést reprezentáló osztály a következő adattagokkal: *prototype* (alakzat típusa, *string*, elfogadott értékek: **Circle**, **Triangle**, **Rectangle**), *amount* (rendelt mennyiség, *int*), *x*, *y* (a fenti alakzatoknál bemutatott plusz adattagok, **Circle** esetén az *y* = 0, *int*). Van továbbá egy *print* metódusa, ami kiírja a rendelés néhány adatát.

Prototípusok kezelése

Az első részfeladat az alakzat osztályok felkészítése a klónozásra, illetve a klónozásra kerülő prototípus objektumok tárolásának és kezelésének megvalósítása.

1. A **Shape** osztálynak van egy tisztán virtuális *Clone* metódusa, ami a klónozást végzi. A származtatott osztályokban valósítsd meg a *Clone* függvényt a klónozás megfelelő működéséhez (**Clone** minta), (3 pont)
2. készíts egy **ShapeFactory** osztály, ami a prototípusokat tárolja és kezeli: legyen egy *map* adattagja, aminek a kulcsa az alakzat típus (**Circle**, **Triangle**, **Rectangle**) értéke pedig **Shape***, (1 pont)
 - a. a **ShapeFactory** osztály legyen egy *factory* nevű névtérben és ennek megfelelően használd majd **main.cpp**-ben, (2 pont)
 - b. a **ShapeFactory** osztályból csak egy példány létezhesen a **Singleton** mintának megfelelően, (2 pont)
 - c. a destruktornak szabadítsd fel a tárolt prototípusok által dinamikusán foglalt memóriát, (1 pont)
 - d. legyen az osztálynak egy *addPrototype* metódusa, ami egy **Shape*** prototípust kap paraméterként és azt eltárolja az asszociatív tömbben, (1 pont)
 - e. legyen egy *getPrototype* metódusa, ami egy alakzat típust kap paraméterként, megnézi hogy van-e ilyen prototípus a tárolóban, ha van akkor lemásolja és visszatér vele, ha nincs akkor kiír egy hibaüzenetet és null-al tér vissza, (2 pont)
 - f. + a fenti hibaüzenet helyett készíts egy saját kivétel osztályt és azt dobd el ha nem található a keresett típus, a kivétel üzenete tartalmazza a helytelen típus nevét is. A kivétel kezelését egy későbbi pontban leírt függvényben kell megoldani majd. (2 pont)

Prototípusok és rendelések beolvasása

A létrehozandó prototípusokat és a teljesítendő rendeléseket az input.txt fájl tartalmazza. A **PROTOTYPE** kezdetű sorok a prototípusokat, az **ORDER** kezdetű sorok pedig a rendeléseket reprezentálják. A **PROTOTYPE** sorokban a sor típusa után az alakzat típusa van utána a mérete végül a színe. Az **ORDER** kezdetű sorokban pedig a sor típusa után az alakzat típusa következik, utána a rendelt mennyiség végül pedig az alakzat paraméterei (*x, y*, mindig két egész szám, **Circle** esetében a második mindig 0). Minden sor egy-egy objektumot reprezentál, a beolvasás végét a '*' karakter jelzi (a sor típusa helyett), az adatok típusa megfelelő, ellenőrizni nem kell.

3. A **main** függvényben hozz létre egy listát, ami **Order** objektumokat tárol, (1 pont)
4. készíts továbbá a **main.cpp**-ben egy **readInput** nevű függvényt, ami egy fájlnevet és az előbbi listát referenciaként kapja paraméterként majd az alábbiakban részletezett módon beolvassa a fájl tartalmát: (2 pont)
 - a. az **ORDER** típusú sorokból létrehoz egy-egy **Order** objektumot és beszúrja a rendeléseket tároló listába, (1 pont)
 - b. a **PROTOTYPE** típusú sorokból az alakzat típusának megfelelő objektumot kell létrehozni és eltárolni a **ShapeFactory** **addPrototype** metódusával. (3 pont)

Rendelések feldolgozása

5. Szintén a **main.cpp**-be készíts egy **processOrders** függvényt, ami a rendeléseket tartalmazó listát kapja paraméterként és az alábbi módon feldolgozza azokat:
 - a. menj végig a listán **range-based loop** használatával és a példakimenetnek megfelelően jelezd a rendelés feldolgozásának megkezdését. A rendelésben szereplő alakzat típusból hozz létre a rendelésben szereplő mennyiségű másolatot a **ShapeFactory** **getPrototype** metódusával (figyelj, hogy null visszatérési érték is lehet ha nem használtál kivételt), majd a klón alakzat **setParameters** függvényével állítsd be a rendelésben szereplő paramétereket (*x, y*) a klón objektumokban. A klón objektumokat nem használjuk többet, gondoskodj a dinamikusan foglalt memória felszabadításáról, végül a példakimenetnek megfelelően jelezd a feldolgozás befejezését és írd ki az előállított mennyiséget, (4 pont)
 - b. + amennyiben a **getPrototype** függvényben saját kivétel osztályt használtál, akkor itt kezeld az esetleg es kivételt úgy, hogy ha egy rendelés esetében az első **getPrototype** hívás kivételt dob, akkor az üzenet kiírása után a következő rendeléssel folytasd, (1 pont)
 - c. + készíts egy lambda kifejezést, ami egy **Order** objektumot vár paraméterként és az a. pontban leírt módon elvégzi egy rendelés feldolgozását, (2 pont)
 - d. + minden megrendelés feldolgozását egy külön szálon végezd el a fenti lambda kifejezés használatával úgy, hogy minden szál számolja meg, hogy hány alakzatot gyártott le és a feldolgozás után írasd ki a szálak által gyártott összes alakzat számát is, figyelj a szálak megfelelő visszacsatolására a fő szálhoz, (2 pont)
 - e. + a rendelés feldolgozó szálak párhuzamosan végeznek kiírás a konzolra, kölcsönös kizárás segítségével oldd meg, hogy a kiírások szinkronizáltak legyenek (ne keveredjenek a sorok a kimeneten). (2 pont)