



RESIDENT EVIL – CODENAME: NIK

Gyakorló zárthelyi dolgozat

Figyelem!

A feladat mennyiségre több anyagot tartalmaz, mint a zárthelyi dolgozat, amely az órán várható. Ennek oka, hogy több gyakorlásra legyen lehetőség.

Megjegyzés!

Az osztályokban, metódusokban, változóknak NE használjon ékezetes karaktereket. A feladat kiírásában a megértés és olvashatóság elősegítése végett kerültek ékezettel írottak.

A feladathoz megtalálható a teljes kód feltöltve a weboldalamon, elakadás esetén érdemes első körben gondolkodni, második körben pedig megnézni, hogy „mire is gondolt a költő”. Amennyiben a feladat első körre túl nagy falat és „belezavarodok” érzést kelt egy-egy része, akkor javaslom, hogy az órai kisebb mintapéldákat elővenni és ott leellenőrizni a tanultakat.

Megjegyzés!

Amennyiben hibát / ellentmondást / bármi problémát vél felfedezni a leírásban, kérem jelezze azt a sipos.miklos@nik.uni-obuda.hu elérhetőségen.

1. Interfész osztályok

IEllenség

- int életerő G/S
- int sebzés G/S
- támadás metódus IJátékos típusú paraméterrel, logikai visszatéréssel

IFőEllenség

- terjessze ki az IEllenség-et
- int legyőzhetőség G/S

IJátékos

- string név G/S
- int életerő G/S
- int sebzés G/S
- JatekosKarakter karakter G/S
 - o saját enumeráció típus (opciók: Leon, Jill, Chris)
- támadás metódus IEllenség típusú paraméterrel, visszatérés nélkül

IKivételek

- string hibaüzenet G/S

2. Kivétel osztályok

AlapException

- valósítsa meg az IKivételek interfészt

EllenfelekTeleException

- származzon le az AlapException osztályból
- int elemszám G/S

ZombiModellBetoltesException

- származzon le az AlapException osztályból
- Zombi zombi G/S

3. Entitás osztályok

Zombi

- valósítsa meg az IEllenség interfészt
- biztosítsa, hogy az osztályból példányt ne lehessen létrehozni
- rendelkezzen (csak) egy konstruktorral, amely az életerőt fogadja paraméternek
- biztosítsa, hogy az osztályból származtatott elemek a ToString metódust kötelezően felüldefiniálják majd, de itt ebben az osztályban még nem dönthető el, hogy mi legyen a felüldefiniált kimenet
- készítsen hasonló elv mentén egy életerő visszatöltés metódust, amelyet a leszármaztatott osztályok kötelező jelleggel implementálniuk kell majd
- támadás esetén ne tegyen semmit, csak adjon vissza hamis értéket
- hozzon létre egy logzombi virtuális metódust, amely az adott példány típusát írja ki a konzolra

BioZombi

- származzon le a Zombi ősoosztályból
- int sugárzás G/S
- ToString felüldefiniálás esetén adja vissza a következőt, a szükséges helyekre helyettesítse be az értékeket
 - o BIOZOMBI – (életerő) (sebzés) (sugárzás)
- támadás esetén 10-zel sebezze a játékost
- biztosítsa, hogy az osztályból ne lehessen tovább örökíteni
- írja felül az életerő visszatöltésért felelős metódust, amely a zombi egyed életerejét 10-el megnöveli
- írja felül a logzombi metódust, a példány saját típusával
- modell generálás metódus, amely csak az osztályon belül látható és érhető el. Ezen metódus célja, hogy hibát szimulál. 50%-os valószínűsége van a hibának, amely esetén egy ZombiModellBetoltesException kivétel dobódik a rendszerben, hibaüzenettel és egy referenciával a zombira, amely kiváltotta a hibát. Egyéb esetben nem kell mást tenni, a feladatnak nem része a grafikus modell megalkotása.

FutoZombi

- származzon le a Zombi őszosztályból
- int stamina G/S
- ToString felüldefiniálás esetén adja vissza a következőt, a szükséges helyekre helyettesítse be az értékeket
 - o FUTOZOMBI – (életerő) (sebzés) (stamina)
- támadás esetén 15-tel sebezze a játékost
- biztosítsa, hogy az osztályból ne lehessen tovább örökíteni
- írja felül az életerő visszatöltésért felelős metódust, amely a zombi egyed életerejét 10-zel megnöveli
- írja felül a logzombi metódust, a példány saját típusával

Tyrant

- valósítsa meg az IFőEllenség interfészt
- támadás esetén az entitás saját sebzésének értékével sebezze a játékost
- biztosítsa, hogy az osztályból ne lehessen tovább örökíteni

Nemesis

- valósítsa meg az IFőEllenség interfészt
- támadás esetén az entitás saját sebzésének értékével sebezze a játékost
- biztosítsa, hogy az osztályból ne lehessen tovább örökíteni

Játékos

- valósítsa meg az IJátékos interfészt
- támadás metódus esetén az ellenség életerejét csökkentse a sebzés mértékével
- definiálja felül a ToString metódust, név és életerőt adjon vissza

6. Eddigiek tesztelése

Hozzon létre egy Zombi referenciával rendelkező négy elemű tömböt, töltsse fel futó- és biozombi típusú elemekkel. Generáljon az egyes típusoknak véletlenszerűen sebzési értéket, típustól függően pedig stamina és sugárzás értékeket. A zombik modellének létrehozásakor figyeljen a kivételekre, kezelje le azokat a tanultaknak megfelelően. Hívja meg a logzombi metódust, vizsgálja meg a polimorfizmust az objektumok tömbjének minden elemére.

7. Játék osztály

Készítsen egy játék osztályt, amely az egész program lényegét és irányítását fogja adni. A konstruktorban lehessen megadni, hogy a játékon belül hány darab ellenséget tudunk eltárolni. Ehhez használjon tömb adatszerkezetet. Készítsen egy külső változót, amely mindig az aktuális ellenségszámot mutatja majd. Tehát az osztályon belül található elemek:

- Játékos játékos G/S
- IEllenség típusú elemek tárolására alkalmas tömb adatszerkezet G/S
- ellenfélszám G/S
- EllenfélHozzáadás metódus, amely egy IEllenség típusú elemet fogad és a következő helyre helyezi a tömbben
 - o amennyiben nincs már hely, úgy dobjon egy saját készítésű **EllenfelekTeleException** kivételt, hibaüzenetnek adja meg, hogy nincs már több hely ellenfélnek, továbbá azt, hogy hány elemű gyűjtemény (jelenleg tömb) volt az, amely betelt
- EllenfelekKilistázása metódus, amely az ellenfeleket kiírja egymás alá a konzolra
- Támadás metódus, amely egy IEllenség típust fogad és a játékos támadás metódusát hívja tovább, átadva neki a paraméterként kapott ellenséget

8. Eddigiek tesztelése

Hozzon létre egy játék objektumot a Main részben, regisztráljon hozzá játékost is amelyet tetszőlegesen paraméterezzen fel. Adjon hozzá a játékhoz ellenfeleket, amelyeket korábban hozott létre, készítse fel a programot, hogy ezen a részen kivétel dobódhat. Kezelje azt le, és biztosítsa, hogy más kivétel dobása esetén se fagyjon le a program. Listázza ki a játékban található ellenfeleket.

9. Események integrálása

Figyelem!

Az eseményeknél a gyakorlás céljából két verzióban található meg a feladatléírás. A **rövidebb** verzióban minden lényegi rész említésre kerül, az alapján a feladat megvalósítható. A **részletesebb** verzióban bővebben ki vannak fejtve, hogy egyes dolgokat hogyan kellene megvalósítani. Aki magabiztos, próbálja meg a rövidebb leírás alapján. Aki kevésbé, nekik javasolt a részletesebb leírás.

Rövid megfogalmazás:

A játékos osztályban hozzon létre egy delegáltat és egy hozzá tartozó eseményt **TalalatEventHandler** néven, a tanult módon. Amikor a játékban az ellenfelet támadás éri, süsse el az előbbi eseményt. Biztosítsa, hogy az eseménynek legyen saját argumentuma **TalalatEventArgs** néven. Ennek segítségével egy IEllenség típusú ellenséget, egy Játékos típusú játékost és egy DateTime típusú időbélyeget tudunk az eseménnyel küldeni (amelyek mind olvashatók és írhatók).

Hasonló képpen járjon el a Zombi osztályban is, amelynél egy **MeghaltEventHandler** delegált és eseményre lesz szükség. Ezt az eseményt akkor süsse el, ha a zombi életereje nulla vagy az alá csökkent. Készítsen egy **ZombiMeghaltEventArgs** osztályt, amelyben egy IEllenség típust és egy DateTime típust tudunk küldeni az eseménnyel.

Mind a két esemény elsütése előtt vizsgálja meg, hogy elsüthető-e az esemény hibára futás nélkül.

Részletes megfogalmazás:

Hozzon létre a játékos osztályban egy delegáltat **TalalatEventHandler** néven, amely visszatérés nélküli metódusszignatúrát ír elő, egy object és egy **TalalatEventArgs** típusú paraméterrel. Készítsen egy eseményt Talalat néven ehhez a delegálthoz. Módosítsa a támadás metódust úgy, hogy miután az ellenség életerejének csökkentése megtörtént, süsse el az eseményt a tanult módon, melyhez használjon egy OnTalalat metódust. Vizsgálja meg, hogy vannak-e feliratkozva már az eseményre, és csak akkor süsse azt el. Az eseményargumentumon keresztül kommunikálja az ellenséget és a játékost, továbbá az aktuális pillanattal hozzon létre egy időbélyeget.

A TalalatEventArgs osztályt készítse el a következő paraméterekkel:

- IEllenség ellenség G/S
- Játékos játékos G/S
- DateTime időbélyeg G/S

Hozzon létre a zombi osztályban egy delegáltat **MeghaltEventHandler** néven, amely visszatérés nélküli metódusszignatúrát ír elő, egy object és egy **ZombiMeghaltEventArgs** típusú paraméterrel. Készítsen egy eseményt Meghalt néven ehhez a delegálthoz. Módosítsa az életerő set ágát oly módon, hogy miután az új értékkel felülírta az életerőt, vizsgálja meg, hogy az nulla vagy annál kisebb-e. Amennyiben igen, úgy jelezzen esemény formájában, melyhez használja az OnMeghalt nevű metódust. Elsütés előtt vizsgálja meg, hogy van-e feliratkozás az eseményre, és csak akkor süsse el. A ZombiMeghaltEventArgs osztályon keresztül adja át a zombit magát és egy időbélyeget is hozzon létre.

A ZombiMeghaltEventArgs osztály a következő paraméterekkel rendelkezik:

- IEllenség ellenség G/S
- DateTime időbélyeg G/S

Hozzon létre a játék osztályban két figyelőt TalalatFigyeloService és MeghaltFigyeloService neveken, amelyek **JátékTalalatFigyelőService** és **ZombiMeghaltFigyelőService** típusúak. Az első a játékos támadását fogja figyelni, és minden támadáskor logolni fog egy bejegyzést. A második akkor fog esemény formájában logbejegyzést készíteni, ha az egyik zombi meghalt. A játék osztályban készítsen egy EseményFeliratkoztatás metódust, amely a következőket csinálja:

- TalalatFigyeloService példány létrehozása és esemény feliratkoztatása
- MeghaltFigyeloService példány létrehozása és minden ellenfél saját eseményének a feliratkoztatása

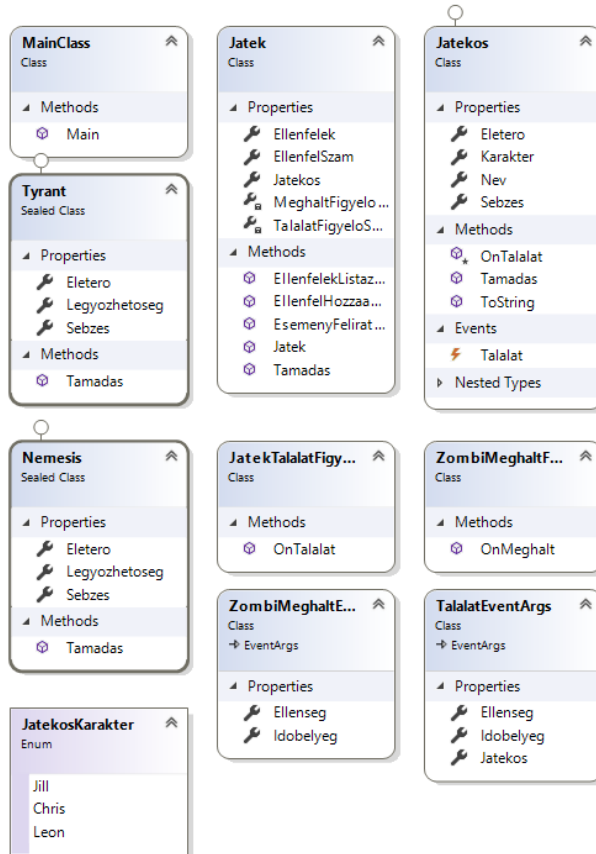
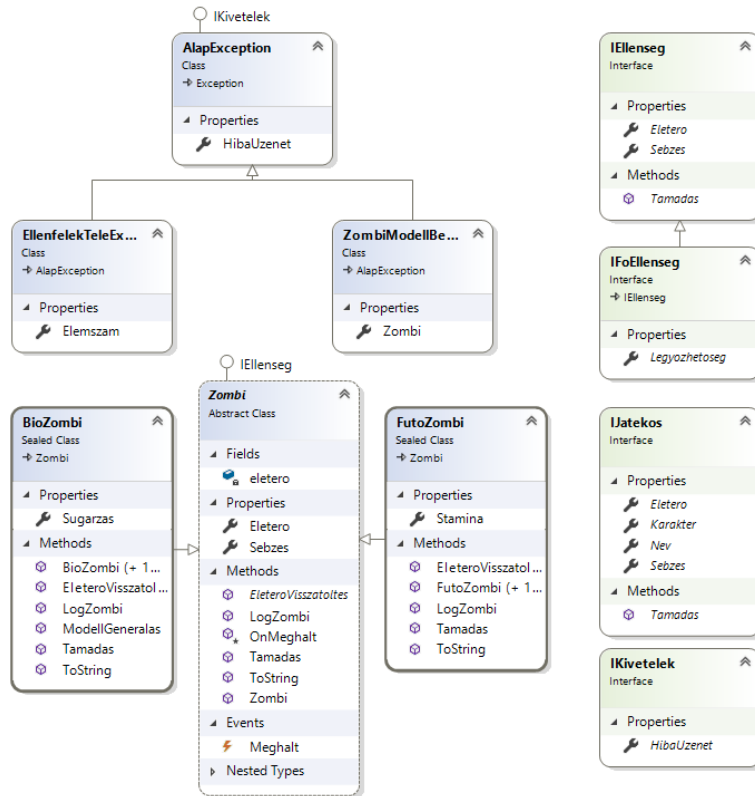
JátékTalalatFigyelőService osztályban hozzon létre egy megfelelő metódusszignatúrával rendelkező metódust, amely kiírja a bejegyzés időpontját, majd azt, hogy melyik játékos melyik ellenfelet találta el, és az ellenfél új életerejét.

ZombiMeghaltFigyelőService osztályban hozzon létre egy megfelelő metódusszignatúrával rendelkező metódust, amely kiírja a bejegyzés időpontját, majd azt, hogy melyik ellenség halt meg.

10. Eddigiek tesztelése

Tesztelje le az eseményeket, támadjon a játékosal egy zombi ellenségre, vizsgálja meg, hogy meghal-e megfelelő mennyiségű támadás után. További kivételek és események elhelyezhetők tetszés szerint, amelyek további gyakorlásra alkalmasak.

Osztályok vizualizálása



A program elvárt kimenete

WELCOME TO RE CODENAME NIK
ERR: modell betöltése sikertelen.
BIOZOMBI - (100) (0) (0)
[LOG]: FUT0: mintazhresidentevilcodenamenik.FutoZombi
[LOG]: FUT0: mintazhresidentevilcodenamenik.FutoZombi
[LOG]: BIO: mintazhresidentevilcodenamenik.BioZombi
[LOG]: BIO: mintazhresidentevilcodenamenik.BioZombi
ERR: Nincs hely több ellenfélnek a játékban.
>FUTOZOMBI - (100) (7) (0)
>FUTOZOMBI - (100) (1) (7)

LOG [2019. 02. 05. 9:16:01]:
A Leon S. Kennedy (100) játékos eltalálta a(z) FUTOZOMBI - (80) (7) (0) ellenfelet.
Ellenfél új életereje: 80

LOG [2019. 02. 05. 9:16:02]:
A Leon S. Kennedy (100) játékos eltalálta a(z) FUTOZOMBI - (60) (7) (0) ellenfelet.
Ellenfél új életereje: 60

LOG [2019. 02. 05. 9:16:02]:
A(z) FUTOZOMBI - (-1) (7) (0) ellenség meghalt.

LOG [2019. 02. 05. 9:16:02]:
A Leon S. Kennedy (100) játékos eltalálta a(z) FUTOZOMBI - (-1) (7) (0) ellenfelet.
Ellenfél új életereje: -1

Press any key to continue...|