



RESIDENT EVIL – CODENAME: NIK

Gyakorló zárthelyi dolgozat

Figyelem!

A feladat feltételezi, hogy minden olyan ismeretanyag elsajátítása megtörtént, amely az 1. zh anyagát képezte. Továbbá, ez a feladat annak a folytatása, így az abban található kódok is szükségesek.

Megjegyzés!

Az osztályokban, metódusokban, változóknál 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. Továbbá, egyes részek [nincs implementálva a kódban ez a rész] megjelölést kaptak. Ez azt jelenti, hogy azon részek nincsenek megírva, mert vagy az órai munka copy-paste változata, vagy a meglévő tudás alapján összerakhatók, ez egyben a feladat része!

Megjegyzés!

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

1. Adatszerkezet osztályok

Készítse el a következő adatszerkezeteket külön osztályokban, ezt követően majd ezek lesznek felhasználva az egyes osztályokban.

LLista

- generikus egyirányú láncolt lista
- elem beszúrási lehetőségek
 - o lista eleje
 - o lista vége
 - o lista n-edik helyére
- elem törlése
- elem módosítása

RLLista

- generikus egyirányú rendezett láncolt lista
- listába beszúrási
- listából törlés
- elem módosítása a listában

IAdatkezelő

- interfész
- int aktív elemszám G/S

ŐsAdatszerkezet

- legyen generikus
- ne lehessen példányt hozni belőle közvetlenül
- metódusok törzs nélkül
 - o elemberek T típust fogad
 - o elemkivesz T típust ad vissza
 - o elemmegtekint T típust ad vissza

TVerem & TSor

- valósítsa meg az IAdatkezelő interfészt
- származzon le az ŐsAdatszerkezetből
- legyen generikus
- belső adatszerkezetnek használjon tömböt
- elem berakása
- elem kivétele
- dobjon kivételeket a következő esetekben
 - o berakás esetén, ha nincs már hely: NincsHelyException
 - o kivétel és megtekintés esetén, ha üres az adatszerkezet: NincsElemException

LVerem & LSor [nincs implementálva a kódban ez a rész]

- valósítsa meg az IAdatkezelő interfészt
- származzon le az ŐsAdatszerkezetből
- legyen generikus
- belső adatszerkezetnek használjon egyirányú láncolt listát (LLista)
- elem berakása
- elem kivétele
- dobjon kivételeket a következő esetekben
 - o berakás esetén, ha nincs már hely: NincsHelyException
 - o kivétel és megtekintés esetén, ha üres az adatszerkezet: NincsElemException

Gráf [nincs implementálva a kódban ez a rész]

- legyen generikus
- szomszédsági listával
- csúcsmátrixszal

BinárisKeresőFa [nincs implementálva a kódban ez a rész]

- legyen generikus
- elem beszúrása
- elem törlése
- biztosítsa, hogy a fába csak IEllenség típusú elemek legyenek behelyezhetők

2. Meglévő osztályok módosítása

A játék osztályt egészítse ki egy saját láncolt lista alapú adatszerkezettel (ez lehet a jelenlegi tömb lecserélése, de lehet attól függetlenül egy másik „tároló” is), amelyben zombi típusú elemeket tudunk eltárolni. Készítsen a játék osztályba egy metódust, amely ezt a listát kezeli, zombi hozzáadásakor a listának a beszúr metódusát hívja tovább. Ennek megfelelően végezze el a további metódusokat is a listával kapcsolatosan. [nincs implementálva a kódban ez a rész]

Egészítse ki a láncolt lista osztályt egy delegálttal és hozzá tartozó eseménnyel, amely minden egyes zombi (a lista szempontjából T típusú elem) felvitelekor esemény formájában értesítést küld (logolja a konzolra), hogy új beszúrás történt. Az egyes eseményekkor jelezze, hogy hány aktív eleme van a láncolt listának, valamint az esemény argumentumán keresztül adja meg, hogy mely zombi került felvitelre. Az ehhez szükséges további osztályokat készítse el a tanultaknak megfelelően. [nincs implementálva a kódban ez a rész]

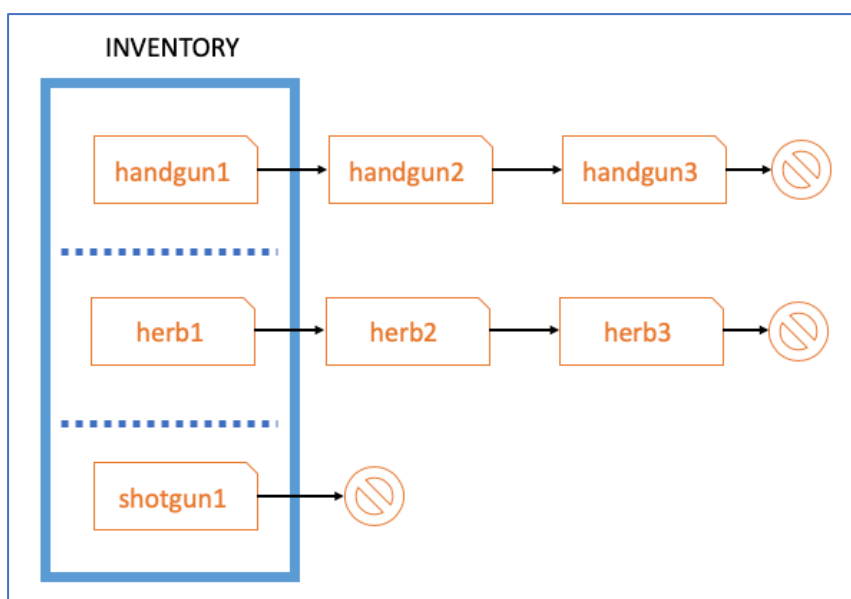
Készítsen egy **FelvehetőElem** osztályt a következő tulajdonságokkal:

- string megnevezés G/S
- datetime felvétel ideje G/S

Származtassa le belőle a következő osztályokat:

- **Kézifegyver**
- **Shotgun**
- **Gyógynövény**
 - o ez rendelkezzen egy string típus G/S tulajdonsággal is (pl. kék, piros, zöld)
- további opciók hozzáadhatók tetszés szerint

Készítsen a játékos osztályba egy verem vagy sor alapú adatszerkezetet, amelynek elemei láncolt listák, amelynek elemei FelvehetőElem típusúak. Ez fogja reprezentálni az inventory-ját a játékosnak. A logika mögöttes az, hogy az inventory kapacitása véges, ahogy a verem mérete is (tömb esetén). A vermen belül egy helyre egy típus fog kerülni, pl. kézfegyver. Ebből azonban lehet több darab is, tehát az újabb felvett darabokat egy láncolt listában kell elhelyezni. Adjon hozzá elemeket, töröljön elemeket, gyakorolja az adatszerkezetekkel való műveleteket.



Vizualizációs ábra az inventory elképzeléséhez.