

Az alábbi feladatok megoldásához több olyan osztályt kell használni, amelyek egy közös ősosztályból származnak és felüldefiniálják az ősosztály virtuális metódusait. Ezen osztályok objektumait egy gyűjteménybe kell elhelyezni, majd ezt a gyűjteményt kell bejárni, a benne levő objektumok megfelelő metódusait meghívni.

1. Készítsünk C++ programot a következő feladat megoldására!

Egy többnapos versenyen lények vesznek részt. **Ki nyeri a versenyt, azaz melyik lény teszi meg a legnagyobb távolságot úgy, hogy közben életben marad?** Kezdetben minden lény valamennyi vízzel rendelkezik, és a megtett távolsága 0. A verseny során háromféle nap lehetséges: napos, felhős és esős. Ezekre a különböző fajtájú lények eltérő módon reagálnak vízfogyasztás és haladás szempontjából. Minden lény először a rendelkezésére álló víz mennyiségét változtatja meg, ezután ha tud, mozog. Bármely lény elpusztul, ha a vize elfogy (0 lesz az érték), ezután értelemszerűen semmilyen tevékenységre sem képes.

Minden lény jellemzői: az egyedi neve (string), a rendelkezésre álló víz mennyisége (egész), a maximálisan tárolható víz mennyisége (egész), hogy él-e (logikai), illetve az eddig megtett távolság (egész). A versenyen részt vevő lények fajtái a következők: homokjáró, szivacs, lépegető.

A következő táblázat tartalmazza az egyes fajták jellemzőit.

| fajta | víz változtatás | | | távolság | | | max.víz |
|-----------|-----------------|--------|------|----------|--------|------|---------|
| | napos | felhős | esős | napos | felhős | esős | |
| homokjáró | -1 | 0 | 3 | 3 | 1 | 0 | 8 |
| szivacs | -4 | -1 | 6 | 0 | 1 | 3 | 20 |
| lépegető | -2 | -1 | 3 | 1 | 2 | 1 | 12 |

Az egyes lények a vízkészlet megváltoztatása során nem léphetik túl a fajtára jellemző maximális értéket, legfeljebb azt érhetik el.

A program egy szövegfájlból olvassa be a verseny adatait! Az első sorban az induló lények száma szerepel. A következő sorok tartalmazzák a lények adatait szóközzel elválasztva: a lény nevét, a fajtáját és a kezdetben rendelkezésére álló víz mennyiségét. A fajtát egy karakter azonosítja: h - homokjáró, s - szivacs, l - lépegető.

A lényeket leíró részt követő sorban a verseny napjai szerepelnek egy karaktorsorozatban. Az egyes jelek értelmezése: n - napos, f - felhős, e - esős.

A program kérje be a fájl nevét, majd jelenítse meg a nyertes nevét! (Feltehetjük, hogy a fájl formátuma helyes.) Egy lehetséges bemenet:

```
4
Vandor h 4
Seta l 7
Csuszo s 12
Siklo s 10
nffeeennf
```

(Javaslat: Hozza létre az absztrakt lény osztályt, amelyből származtatja a homokjáró, szivacs és lépegető osztályokat. A konstruktor paramétere legyen a név és a kezdeti víz mennyisége. Vezessen be három műveletet a napoknak (napos, felhős, esős), amelyek a vízfogyasztást szabályozzák az élet vizsgálatával együtt, illetve mozgatják a lényt. A végeredmény meghatározásához kell még 3 művelet: él-e a lény, a név illetve a megtett távolság lekérdezése.)

2. Készítsünk C++ programot a következő feladat megoldására!

Egy bolygón különböző fajtájú növények élnek, minden növény tápanyagot használ. Ha egy növény tápanyaga elfogy (a mennyisége 0 lesz), a növény elpusztul. A bolygón három fajta sugárzást különböztetünk meg: alfa sugárzás, delta sugárzás, nincs sugárzás. A sugárzásra a különböző fajtájú élő növények eltérő módon reagálnak. A reakció tartalmazza a tápanyag változását, illetve a következő napi sugárzás befolyásolását. A másnapi sugárzás alakulása: ha az alfa sugárzásra beérkezett igények összege legalább hárommal meghaladja a delta sugárzás igényeinek összegét, akkor alfa sugárzás lesz; ha a delta sugárzásra igaz ugyanez, akkor delta sugárzás lesz; ha a két igény közti eltérés háromnál kisebb, akkor nincs sugárzás. Az első nap sugárzás nélküli.

Szimuláljuk a növények viselkedését és minden lépésben írjuk ki az összes növényt a rájuk jellemző tulajdonságokkal, valamint az aktuális sugárzást!

Minden növény jellemzői: az egyedi neve (string), a rendelkezésre álló tápanyag mennyisége (egész), hogy él-e (logikai). A szimulációban részt vevő növények fajtái a következők: puffancs, deltafa, parabokor. A következőkben megadjuk, hogy az egyes fajták miként reagálnak a különböző sugárzásokra. Először a tápanyag változik, és ha a növény ezután él, akkor befolyásolhatja a sugárzást.

Puffancs: Alfa sugárzás hatására a tápanyag mennyisége kettővel nő, sugárzás mentes napon a tápanyag eggyel csökken, delta sugárzás esetén a tápanyag kettővel csökken. Minden esetben úgy befolyásolja a másnapi sugárzást, hogy 10 - tápanyag értékben növeli az alfa sugárzás bekövetkezését. Ez a fajta akkor is elpusztul, ha a tápanyag mennyisége 10 fölé emelkedik.

Deltafa: Alfa sugárzás hatására a tápanyag mennyisége hárommal csökken, sugárzás nélküli napon a tápanyag eggyel csökken, delta sugárzás hatására a tápanyag négyvel nő. Ha a tápanyag mennyisége 5-nél kisebb, akkor 4 értékben növeli a delta sugárzás bekövetkezését, ha 5 és 10 közé esik, akkor 1 értékben növeli a delta sugárzás bekövetkezését, ha 10-nél több, akkor nem befolyásolja a másnapi sugárzást.

Parabokor: Akár alfa, akár delta sugárzás hatására a tápanyag mennyisége eggyel nő. Sugárzás nélküli napon a tápanyag eggyel csökken. A másnapi sugárzást nem befolyásolja.

A program egy szövegfájlból olvassa be a szimuláció adatait! Az első sorban a növények száma szerepel. A következő sorok tartalmazzák a növények adatait szóközzel elválasztva: a növény nevét, a fajtáját és a kezdetben rendelkezésre álló tápanyag mennyiségét. A fajtát egy karakter azonosítja: a - puffancs, d - deltafa, p - parabokor. A növényeket leíró részt követő sorban a szimuláció napjainak száma adott egész számként. A program kérje be a fájl nevét, majd jelenítse meg a túlélők nevét! (Feltehetjük, hogy a fájl formátuma helyes.) Egy lehetséges bemenet:

4

Falánk a 7

Sudár d 5

Köpcös p 4

Nyúlánk d 3

10

(Javaslat: Hozza létre az absztrakt növény osztályt, amelyből az egyes fajták osztályait. A konstruktor paramétere legyen a név és a tápanyag kezdeti mennyisége. Egy nap eltérése is vezessen be művelet(ek)et, a visszatérő érték adja meg, hogy a növény milyen módon befolyásolja a következő nap sugárzását. A befolyásolás lehet egy egész szám, ami pozitív alfa sugárzás esetén, negatív delta sugárzás esetén, nulla, ha a növény nem befolyásolja a másnapi sugárzást. A végeredmény meghatározásához kell még 2 művelet: él-e a lény, illetve a név lekérdezése.)

3. Készítsünk C++ programot a következő feladat megoldására!

Szimuláljuk az alábbi egyszerűsített Capitaly társasjátékot! Adott néhány eltérő stratégiájú játékos és egy körpálya, amelyen különféle mezők sorakoznak egymás után. A pályát körbe-körbe újra és újra bejárják a játékosok úgy, hogy egy kockával dobva mindig annyit lépnek, amennyit a kocka mutat. A mezők három félek lehetnek: ingatlanok, szolgáltatások és szerencse mezők. Az ingatlant meg lehet vásárolni 1000 Petákért, majd újra rálépve házat is lehet rá építeni 4000 Petákért. Ha ezután más játékos erre a mezőre lép, akkor a mező tulajdonosának fizet: ha még nincs rajta ház, akkor 500 Petákot, ha van rajta ház, akkor 2000 Petákot. A szolgáltatás mezőre lépve a banknak kell befizetni a mező paramétereként megadott összeget. A szerencse mezőre lépve a mező paramétereként megadott összegű pénzt kap a játékos. Háromféle stratégiájú játékos vesz részt a játékban. Kezdetben mindenki kap egy induló tőkét (10000 Peták), majd A „mohó” játékos ha egy még gazdátlan ingatlan mezőjére lépett, vagy övé az ingatlan, de még nincs rajta ház, továbbá van elég tőkéje, akkor vásárol. Az „óvatos” játékos egy körben csak a tőkéjének a felét vásárolja el, a „taktikus” játékos minden második vásárlási lehetőséget kihagyja. Ha egy játékosnak fizetnie kell, de nincs elegendő pénze, akkor kiesik a játékból, házai elvesznek, ingatlanjai megvásárolhatókká válnak.

A játék paramétereit egy szövegfájlból olvassuk be. Ez megadja a pálya hosszát, majd a pálya egyes mezőit. Minden mezőről megadjuk annak típusát, illetve ha szolgáltatás vagy szerencse mező, akkor annak pénzdíját. Ezt követően a fájl megmutatja a játékosok számát, majd sorban minden játékos nevét és stratégiáját. A tesztelhetőséghez fel kell készíteni a megoldó programot olyan szövegfájl feldolgozására is, amely előre rögzített módon tartalmazza a kockadobások eredményét.

Írjuk ki, hogy adott számú kör után hogyan állnak (mennyi a tőkéjük, milyen ingatlanokat birtokolnak) a versenyzők!

4. Készítsünk C++ programot a következő feladat megoldására!

Szimuláljuk az alábbi egyszerűsített Capitaly társasjátékot! Adott néhány eltérő stratégiájú játékos és egy körpálya, amelyen különféle mezők sorakoznak egymás után. A pályát körbe-körbe újra és újra bejárják a játékosok úgy, hogy egy kockával dobva mindig annyit lépnek, amennyit a kocka mutat. A mezők három félék lehetnek: ingatlanok, szolgáltatások és szerencse mezők. Az ingatlant meg lehet vásárolni 1000 Petákért, majd újra rálépve házat is lehet rá építeni 4000 Petákért. Ha ezután más játékos erre a mezőre lép, akkor a mező tulajdonosának fizet: ha még nincs rajta ház, akkor 500 Petákot, ha van rajta ház, akkor 2000 Petákot. A szolgáltatás mezőre lépve a banknak kell befizetni a mező paramétereként megadott összeget. A szerencse mezőre lépve a mező paramétereként megadott összegű pénzt kap a játékos. Háromféle stratégiájú játékos vesz részt a játékban. Kezdetben mindenki kap egy induló tőkét (10000 Peták), majd A „mohó” játékos ha egy még gazdátlan ingatlan mezőjére lépett, vagy övé az ingatlan, de még nincs rajta ház, továbbá van elég tőkéje, akkor vásárol. Az „óvatos” játékos egy körben csak a tőkéjének a felét vásárolja el, a „taktikus” játékos minden második vásárlási lehetőséget kihagyja. Ha egy játékosnak fizetnie kell, de nincs elegendő pénze, akkor kiesik a játékból, házai elvesznek, ingatlanjai megvásárolhatókká válnak.

A játék paramétereit egy szövegfájlból olvassuk be. Ez megadja a pálya hosszát, majd a pálya egyes mezőit. Minden mezőről megadjuk annak típusát, illetve ha szolgáltatás vagy szerencse mező, akkor annak pénzdíját. Ezt követően a fájl megmutatja a játékosok számát, majd sorban minden játékos nevét és stratégiáját. A tesztelhetőséghez fel kell készíteni a megoldó programot olyan szövegfájl feldolgozására is, amely előre rögzített módon tartalmazza a kockadobások eredményét.

Írjuk ki, hogy adott számú kör után mely versenyzők birtokolják az egyes ingatlanokat!

5. Készítsünk C++ programot a következő feladat megoldására!

Szimuláljuk az alábbi egyszerűsített Capitaly társasjátékot! Adott néhány eltérő stratégiájú játékos és egy körpálya, amelyen különféle mezők sorakoznak egymás után. A pályát körbe-körbe újra és újra bejárják a játékosok úgy, hogy egy kockával dobva mindig annyit lépnek, amennyit a kocka mutat. A mezők három félék lehetnek: ingatlanok, szolgáltatások és szerencse mezők. Az ingatlant meg lehet vásárolni 1000 Petákért, majd újra rálépve házat is lehet rá építeni 4000 Petákért. Ha ezután más játékos erre a mezőre lép, akkor a mező tulajdonosának fizet: ha még nincs rajta ház, akkor 500 Petákot, ha van rajta ház, akkor 2000 Petákot. A szolgáltatás mezőre lépve a banknak kell befizetni a mező paramétereként megadott összeget. A szerencse mezőre lépve a mező paramétereként megadott összegű pénzt kap a játékos. Háromféle stratégiájú játékos vesz részt a játékban. Kezdetben mindenki kap egy induló tőkét (10000 Peták), majd A „mohó” játékos ha egy még gazdátlan ingatlan mezőjére lépett, vagy övé az ingatlan, de még nincs rajta ház, továbbá van elég tőkéje, akkor vásárol. Az „óvatos” játékos egy körben csak a tőkéjének a felét vásárolja el, a „taktikus” játékos minden második vásárlási lehetőséget kihagyja. Ha egy játékosnak fizetnie kell, de nincs elegendő pénze, akkor kiesik a játékból, házai elvesznek, ingatlanjai megvásárolhatókká válnak.

A játék paramétereit egy szövegfájlból olvassuk be. Ez megadja a pálya hosszát, majd a pálya egyes mezőit. Minden mezőről megadjuk annak típusát, illetve ha szolgáltatás vagy szerencse mező, akkor annak pénzdíját. Ezt követően a fájl megmutatja a játékosok számát, majd sorban minden játékos nevét és stratégiáját. A tesztelhetőséghez fel kell

készíteni a megoldó programot olyan szövegfájl feldolgozására is, amely előre rögzített módon tartalmazza a kockadobások eredményét.

Adjuk meg, melyik játékos nyeri meg a játékot és mekkora vagyona (mennyi a tőkéje, milyen ingatlanokat birtokol) van ekkor!

6. Készítsünk C++ programot a következő feladat megoldására!

Szimuláljuk az alábbi egyszerűsített Capitaly társasjátékot! Adott néhány eltérő stratégiájú játékos és egy körpálya, amelyen különféle mezők sorakoznak egymás után. A pályát körbe-körbe újra és újra bejárák a játékosok úgy, hogy egy kockával dobva mindig annyit lépnek, amennyit a kocka mutat. A mezők három félek lehetnek: ingatlanok, szolgáltatások és szerencse mezők. Az ingatlant meg lehet vásárolni 1000 Petákért, majd újra rálépve házat is lehet rá építeni 4000 Petákért. Ha ezután más játékos erre a mezőre lép, akkor a mező tulajdonosának fizet: ha még nincs rajta ház, akkor 500 Petákot, ha van rajta ház, akkor 2000 Petákot. A szolgáltatás mezőre lépve a banknak kell befizetni a mező paramétereként megadott összeget. A szerencse mezőre lépve a mező paramétereként megadott összegű pénzt kap a játékos. Háromféle stratégiájú játékos vesz részt a játékban. Kezdetben mindenki kap egy induló tőkét (10000 Peták), majd A „mohó” játékos ha egy még gazdátlan ingatlan mezőjére lépett, vagy övé az ingatlan, de még nincs rajta ház, továbbá van elég tőkéje, akkor vásárol. Az „óvatos” játékos egy körben csak a tőkéjének a felét vásárolja el, a „taktikus” játékos minden második vásárlási lehetőséget kihagyja. Ha egy játékosnak fizetnie kell, de nincs elegendő pénze, akkor kiesik a játékból, házai elvesznek, ingatlanjai megvásárolhatókká válnak.

A játék paramétereit egy szövegfájlból olvassuk be. Ez megadja a pálya hosszát, majd a pálya egyes mezőit. Minden mezőről megadjuk annak típusát, illetve ha szolgáltatás vagy szerencse mező, akkor annak pénzdíját. Ezt követően a fájl megmutatja a játékosok számát, majd sorban minden játékos nevét és stratégiáját. A tesztelőséghez fel kell készíteni a megoldó programot olyan szövegfájl feldolgozására is, amely előre rögzített módon tartalmazza a kockadobások eredményét.

Írjuk ki melyik játékos esik ki másodszorra a játékból!

7. Készítsünk C++ programot a következő feladat megoldására!

A hobbi állatoknak az életkedvük megőrzéséhez a táplálékon túl egyéb dolgokra is szükségük van: a halaknak oxigén dús, megfelelő hőmérsékletű vízre; a madaraknak tágas, tiszta kalitkára; a kutyáknak rendszeres foglalkoztatásra. Pisti számos hobbi állatot tart: halakat, madarakat és kutyákat. Állatainak van neve és ismerhető az életkedvüket mutató 0 és 100 között szám (0 esetén az állat elpusztul). Pistinek vannak jobb és rosszabb napjai. Mikor nagyon jó kedvű, egyik állatáról sem feledkezik meg: ilyenkor a halak életkedve 1-gyel, a madaraké 2-vel, a kutyáké 3-mal nő. Átlagos napokon csak a kutyáival foglalkozik, a többi állat életkedve ilyenkor csökken: a halaké 3-mal, a madaraké 1-gyel. Amikor rosszkedvű, csak a legszükségesebb teendőket látja el és ezért minden állat egy kicsit szomorúbb lesz: a halak 5 egységgel, a madarak 3-mal, a kutyák 10-zel.

Az állatok adatait egy szöveges állományban találjuk. Az első sor tartalmazza az állatok számát, amelyet külön-külön sorban az állatok adatai követnek. Ebben egy karakter azonosítja az állat fajtáját (H – hal, M – madár, K – kutya), amit szóköz után az állat neve követ, majd újabb szóköz után a kezdeti életkedve. Az állományban az állatok felsorolását

követő utolsó sorban egy betű sorozat (sztring) írja le Pisti kedvének az egymás utáni napokon való alakulása: j – jó kedvű, a – átlagos, r – rosszkedvű. Feltehetjük, hogy a fájl formátuma helyes.

Szimuláljuk az állatok életkedvének változását Pisti kedvének alakulása során és írja ki az állatok adatait minden nap végén!

8. Készítsünk C++ programot a következő feladat megoldására!

A hobbi állatoknak az életkedvük megőrzéséhez a táplálékon túl egyéb dolgokra is szükségük van: a halaknak oxigén dús, megfelelő hőmérsékletű vízre; a madaraknak tágas, tiszta kalitkára; a kutyáknak rendszeres foglalkoztatásra. Pisti számos hobbi állatot tart: halakat, madarakat és kutyákat. Állatainak van neve és ismerhető az életkedvüket mutató 0 és 100 között szám (0 esetén az állat elpusztul). Pistinek vannak jobb és rosszabb napjai. Mikor nagyon jó kedvű, egyik állatáról sem feledkezik meg: ilyenkor a halak életkedve 1-gyel, a madaraké 2-vel, a kutyáké 3-mal nő. Átlagos napokon csak a kutyáival foglalkozik, a többi állat életkedve ilyenkor csökken: a halaké 3-mal, a madaraké 1-gyel. Amikor rosszkedvű, csak a legszükségesebb teendőket látja el és ezért minden állat egy kicsit szomorúbb lesz: a halak 5 egységgel, a madarak 3-mal, a kutyák 10-zel.

Az állatok adatait egy szöveges állományban találjuk. Az első sor tartalmazza az állatok számát, amelyet külön-külön sorban az állatok adatai követnek. Ebben egy karakter azonosítja az állat fajtáját (H – hal, M – madár, K – kutya), amit szóköz után az állat neve követ, majd újabb szóköz után a kezdeti életkedve. Az állományban az állatok felsorolását követő utolsó sorban egy betű sorozat (sztring) írja le Pisti kedvének az egymás utáni napokon való alakulása: j – jó kedvű, a – átlagos, r – rosszkedvű. Feltehetjük, hogy a fájl formátuma helyes.

Nevezze meg a legszomorúbb (legkisebb az életkedvű) állatot, amelyik még nem pusztult el a vizsgált napok után! Ha több ilyen életkedvű állat is létezik, akkor írja ki az összesnek a nevét!

9. Készítsünk C++ programot a következő feladat megoldására!

Kati házi kedvencei a tarantulák, az arany hörcsögök és a macskák. Az állatoknak az életerejük megőrzéséhez a táplálékon túl egyéb dolgokra is szükségük van: a tarantuláknak száraz és meleg terráriumra; az arany hörcsögöknek puha alomra, ahová befürhatják magukat, a macskáknak rendszeres simogatásra. Kati állatainak van neve és ismerhető az életerejüket mutató 0 és 70 között szám (0 esetén az állat elpusztul). Katinak vannak jobb és rosszabb napjai. Mikor vidám, egyik állatáról sem feledkezik meg: ilyenkor a tarantulák életereje 1-gyel, a hörcsögöké 2-vel, a macskáké 3-mal nő. Átlagos napokon csak a macskáival foglalkozik (életerejük 3-mal nő), a többi állat életereje ilyenkor csökken: a tarantuláké 2-vel, a hörcsögöké 3-mal. Amikor szomorú, csak a legszükségesebb teendőket látja el és ezért minden állat egy kicsit gyengébb lesz: a tarantulák 3 egységgel, a hörcsögök 5-tel, a macskák 7-tel.

Az állatok adatait egy szöveges állományban találjuk. Az első sor tartalmazza az állatok számát, amelyet külön-külön sorban az állatok adatai követnek. Ebben egy karakter azonosítja az állat fajtáját (T – tarantula, H – hörcsög, M – macska), amit szóköz után az állat neve követ, majd újabb szóköz után a kezdeti életereje. Az állományban az állatok felsorolását követő utolsó sorban egy betű sorozat (sztring) írja le Kati kedvének az

egymás utáni napokon való alakulása: v – vidám, a – átlagos, s – szomorú. Feltehetjük, hogy a fájl formátuma helyes.

Szimulálja az állatok életkedvének változását Kati hangulatváltozásainak nyomán és írja ki az állatok adatait minden nap végén!

10. Készítsünk C++ programot a következő feladat megoldására!

Kati házi kedvencei a tarantulák, az arany hörcsögök és a macskák. Az állatoknak az életerejük megőrzéséhez a táplálékon túl egyéb dolgokra is szükségük van: a tarantuláknak száraz és meleg terráriumra; az arany hörcsögöknek puha alomra, ahová befúrhatják magukat, a macskáknak rendszeres simogatásra. Kati állatainak van neve és ismerhető az életerejüket mutató 0 és 70 között szám (0 esetén az állat elpusztul). Katinak vannak jobb és rosszabb napjai. Mikor vidám, egyik állatáról sem feledkezik meg: ilyenkor a tarantula életereje 1-gyel, a hörcsögöké 2-vel, a macskáké 3-mal nő. Átlagos napokon csak a macskáival foglalkozik (életerejük 3-mal nő), a többi állat életereje ilyenkor csökken: a tarantuláké 2-vel, a hörcsögöké 3-mal. Amikor szomorú, csak a legszükségesebb teendőket látja el és ezért minden állat egy kicsit gyengébb lesz: a tarantula 3 egységgel, a hörcsögök 5-tel, a macskák 7-tel.

Az állatok adatait egy szöveges állományban találjuk. Az első sor tartalmazza az állatok számát, amelyet külön-külön sorban az állatok adatai követnek. Ebben egy karakter azonosítja az állat fajtáját (T – tarantula, H – hörcsög, M – macska), amit szóköz után az állat neve követ, majd újabb szóköz után a kezdeti életereje. Az állományban az állatok felsorolását követő utolsó sorban egy betű sorozat (sztring) írja le Kati kedvének az egymás utáni napokon való alakulása: v – vidám, a – átlagos, s – szomorú. Feltehetjük, hogy a fájl formátuma helyes.

Listázza ki, hogy az egyes napok végén melyik volt a legéletrevalóbb állat!