

12.bc programozás beadandó

Információk

A beadandó négy részből fog állni. Programozás konzolon és grafikus felületen, dokumentálás, tesztelés. Mindegyik feladatrészt egy-egy jegyet fognak kapni, a megfelelő tantárgynál. Minden projekt tartalmaz egy leírást, ami segíthet a feladat megoldásában. Kezdje a dokumentum összeállításával és vele párhuzamosan állítsa össze a konzolos alkalmazást. Majd írja meg rá a teszteket és készítsen hozzá egy grafikus felületet, ami megválaszolja a kérdéseket!

Határidők:

Konzolos alkalmazás + dokumentáció: 2025. március 3.

GUI + Tesztelés: 2025. március 10.

Programozás

1. A projekt készítése során használnia kell öröklődést, kötelező interface-ekkel és osztályokkal. Ha egy szerkezet megengedi, használjon absztrakt osztályt is!
2. Minden projektnél gondolja át milyen interface-ekkel és osztályokkal szeretne dolgozni. Szedje össze és dokumentálja le! Mindenhez írjon magyarázatot! Dokumentálnia kell az elkészített osztályokat, metódusokat is (Summary ///)! A kód dokumentálását elég a projekt végén elvégezni!
3. Az osztályok megírása után, készítsen szöveges állomány(oka)t, amivel a feladat megoldása során dolgozik (*.txt, *.csv ...)!
4. Kezelje le minden esetben az adatok valid tulajdonságát!
 - a. A szövegnél a null-e, 0 hosszú-e és valamennyi karakterből áll-e!
 - b. A számnál várható intervallumon belül van-e!
5. Készítsen a konzolos felületen egy futtatható osztályt, ami válaszol a feladatban megadott kérdésekre! (Adatot gyűjt ki, adatot dolgoz fel, módosít stb.)
 - a. Használjon kivételkezelést (Try-catch)! Figyeljen rá, hogy a program nem állhat le hibával!
6. Mindenhol olyan metódussal dolgozzon, amit le is tud tesztelni!
7. Minden metódus mélysége legfeljebb 2 lehet! Legyen minden kiszervezve külön metódusba.
8. Figyeljen a tiszta kód elveire:
 - a. Egy metódus hossza (zárójelek nélkül) 8-12 sor lehet.
 - b. Minden legyen kiszervezve metódusokba.

9. A konzolos, tesztelési feladat részek után készítsen egy grafikus felületet, ahol az adatokat felhasználja!
 - a. Szükséges a rajz vagy grafikon vagy táblázat stb. használata. Amit a feladat kér/ajánl.
 - b. Amelyik funkció megköveteli az azonnali futtatást, akkor az ne legyen kipakolva külön gomb lenyomásra!
 - c. A grafikus felületnek esztétikusnak és az eseményeket tekintve jól működőnek kell lennie.
 - d. Nem működő funkciót ne építsen be! Pontlevonás jár érte!

Tesztelés:

1. minden metódus rendelkezzen Summary-val (///)
2. minden metódus rendelkezzen tesztfüggvénnyel, a tesztfüggvény közvetlenül a metódushoz legyen megírva,
3. minden adattaghoz használt eljárás is rendelkezzen tesztfüggvénnyel,
4. nem kell minden függvényhez kimerítő tesztet írni, de elvárás, ,
5. szerepeltesse az összes tanult attribútumot,
6. használjon legalább 1 olyan attribútumot is, amiről tanórán nem esett szó.
7. Extra pont a kivételkezelés teszteléséért! (Max +5 pont)

Dokumentáció:

1. dokumentálja az interface-eket, osztályokat, ad mindegyikhez egy leírást!
2. bemutatja a főbb funkciókat. Kifejti a program megírása közben létrehozott főbb osztályok lényegét, folyamatát.
3. bemutat legalább 10 különálló tesztet, ami különböző nézőpontból nézi meg az adott metódust.
4. bemutatja a program működését, képekkel illusztrálva, pontos feladat leírással!
5. Formázásnál vegye figyelembe a *Dokumentáció értékelés főbb szempontjai.pdf* tartalmát!

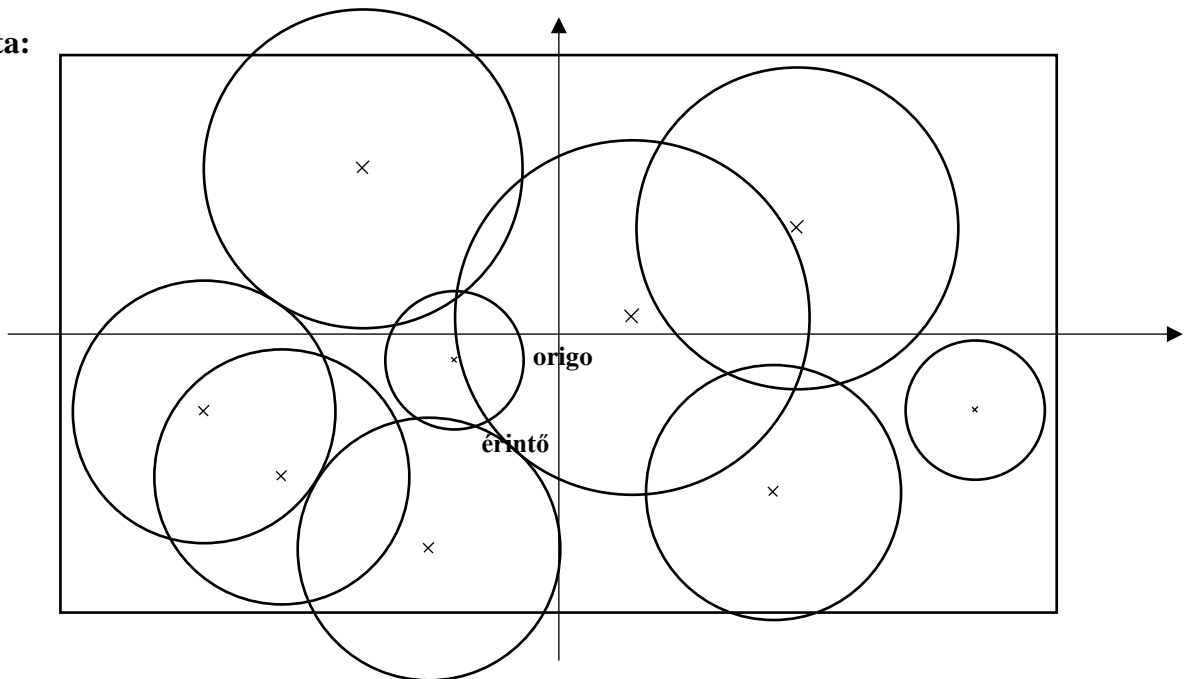
Körök

1. **Projekt:** Írjon programot, amely listában tárolt körökkel statisztikákat és számításokat végez!
 - a. Adott egy vagy több szöveges állomány, amit Önnek kell létrehoznia a feladat leírás alapján! Az adatok beolvasása során a beolvasott adatokra adja meg a következő választ!
 - b. **Kör**
 - Adott a középpontjának az „x” és „y” koordinátái, a harmadik szám a kör sugara „r”.
 - Lekérhető legyen a kör Területe, Kerülete és az origótól mért Távolsága (középpontra nézve)
 - Tudja megadni egy másik körrel van-e metszés- vagy érintéspontja
 - Tudja megadni egy másik körrel való távolságát
 - Lehessen a kör adatait egyben lekérni.
 - c. **Statisztika**
 - Tárolja el egy egységes adatszerkezetben a körök adatait!
 - Ha nincs beolvasott fájl, ne álljon le hibával, ajánljon fel egy generálási lehetőséget a felhasználónak, ami a lehetőségekhez mérten generálja ki a körök adatait. A felhasználó adhassa meg a darabszámot, ha annyit nem tud kigenerálni, akkor figyelmeztessen a program és ajánljon egy darabszámot! Figyeljen rá, hogy megjelenítés során legfeljebb 20%-os mértékben csúszhatnak össze a körök!
 - Válaszoljon a következő kérdésekre:
 1. Lehessen kiírni az összes kör adatát!
 2. Lehessen lekérni, hogy hány darab kör tartalmazza az origót.
 3. A körök mekkora területet foglalnak el a megjelenítő felületen. Ha a körök összecsúsznak, akkor a metszetet ne számolja kétszer!
 4. Lehessen lekérni, hogy melyik kör van a legtávolabb!
 5. Lehessen lekérni két kör távolságát!
 6. Lehessen megkeresni azokat a köröket, amelyek csak érintik egymást!
 - d. Készítsen a feladathoz egy grafikus megjelenítő alkalmazást! Jelenítse meg a koordináta rendszer tengelyeit és körök középpontját és magukat a köröket! A

megoldás során arányosan alakítsa ki grafikus felület nagyságát, hogy az összes kör arányosan beleférjen és látható legyen!

- e. Hozzon létre egy rajzos felületet, amin ábrázolja a köröket! Legyenek tengelyek, és legyen jelölve a kör középpontja is! Ne legyenek a körök kitöltve!

Minta:



Autószerelő műhely munkabeosztása

2. **Projekt:** Írjon programot, amely egy autószerelő műhely szerelőinek napi munkabeosztási tábláját tervezi meg és jeleníti meg a bejövő munka-megrendeléseknek megfelelően! Adott egy Szerviz, ahol szerelők dolgoznak. Mindegyik szerelőnek megvan a maga munkája. Egy beosztást több szerelő is végezhet, de mindegyik szerelőnek egy beosztása lehet csak! Lehessen több napra (akár dátum alapján) foglalásokat indítani!
- a. Adott egy vagy több szöveges állomány, amit Önnek kell létrehoznia a feladat leírás alapján! Az adatok beolvasása során a beolvasott adatokra adja meg a következő választ!
- b. A Szerelők:
- munkaköre/beosztás: f – futómű; o – olajcsere; e – elektronika; k - kerékcseré
 - nev: a szerelő neve
 - minden szerelő munkába áll és nincs szabadságon 10 órás munkaideje van.
 - Lehessen lekérni a szerelők elfoglaltságát!
 - Lehessen lekérni, hogy hány és melyik munkaórái vannak lefoglalva.
- c. Munkatábla:
- Egy összetett adatszerkezetben tárolja a szerelők adatait!
 - Lehessen lekérni egy beosztás alapján, hogy kik végzik el.
 - Lehessen lekérni egy munkakör alapján, hogy kinél melyik órában van szabad hely!
 - Lehessen rendezni az adatokat név és foglaltság alapján!
 - Lehessen foglalni időpontot. A foglalás kezdő és befejező óra megadásával lehessen. *Pl.: e 2 3, azaz elektronikát szeretnénk javítani a 2. órától az 4. óráig.* Foglalásnál vegye figyelembe, hogy csak szabad helyre lehessen foglalni.
 - Lehessen lekérni egyénileg is és összesen is a szabad órákat!
- d. Hozzon létre egy szerelők.txt fájlt a fentebb megadott formátumban úgy, hogy legalább 10 szerelő szerepeljen benne, és legyen benne olyan is akinek több keresztnéve is van.
- e. Készítsen a feladathoz egy grafikus megjelenítő alkalmazást!
- Jelenítse meg a dolgozókat és az órákat!
 - Lehessen napokhoz kötni a foglalást. Vagy dátum alapján vagy legalább 10 napos intervallumra.

- A megjelenítés során figyeljen a színezésre, illetve a szövegnél a minta alapján a név mellette jelenjen meg a munkaköre.
- Kijelöléssel tudjon foglalni az üres helyekre. A pirosat, azaz a már foglalt helyeket ne tudja felülírni.
- Lehesen lekérni, hogy melyik szerelőnek melyik napon van szabad ideje. Ezt is jelenítse meg valamilyen formában!
- Lehesen az aktuális állapotot elmenteni és következő betöltésnél a megváltoztatott adatsort használni!

f. Extra, az Ön által kitalált funkciókat is építhet bele, ha a fentiek elkészültek!

Példa:

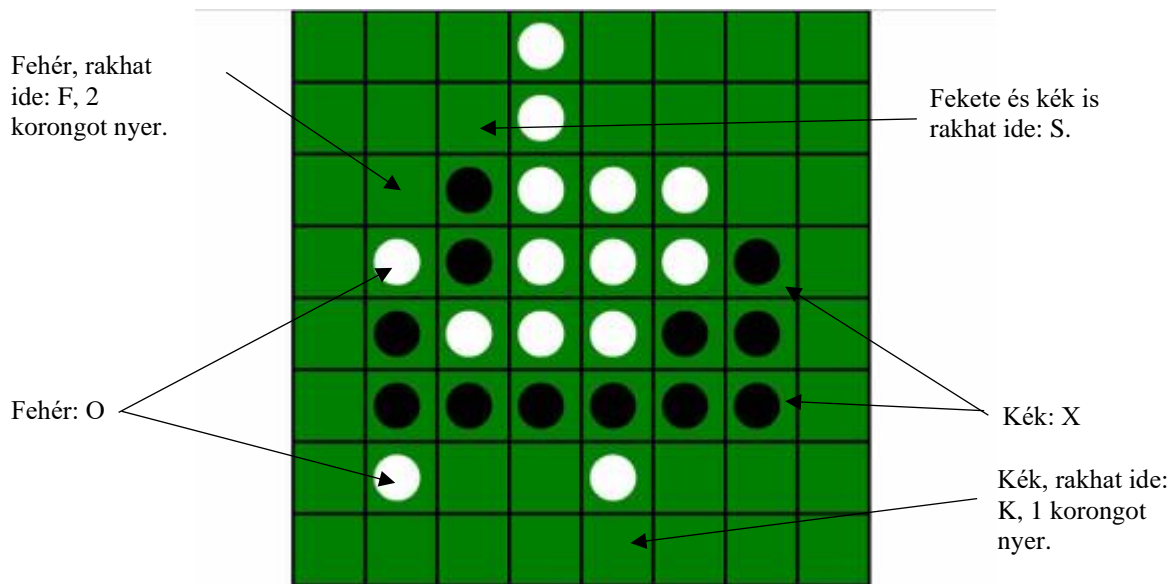
	1	2	3	4	5	6	7	8	9	10
Kis Pista (Olajcsere)										
Banános Joe (Olajcsere)										
Bevíz Elek (Elektronika)										

Olajcsere, foglal 2,2 ➔ Kis Pista

Elektronika, foglal 3,3 ➔ nincs időpont

Reversi

3. **Projekt:** Írjon programot, amely a Reversi játék egy állapotát elemzi ki.
- Hozzon létre egy grafikus felületet, amin ábrázolj a programot!
 - Lehessen látni az aktuális állást!
 - A végén jelezze ki nyert!
 - Lehessen több kört is játszani!
 - Lehessen gép ellen is játszani!
 - Két játékos esetén lehessen menteni az adott állást és visszakérni a mentés eredményét!



Forma 1

4. **Projekt:** Írjon programot, amely a Froma 1 versenytadatait vizsgálja.
- Adott egy vagy több szöveges állomány, amit Önnek kell létrehoznia a feladat leírás alapján! Az adatok beolvasása során a beolvasott adatokra adja meg a következő választ!
 - Dolgozzon a következő adatokkal:
 - Versenyző: neve, kora, magassága, autóinak listája, adott pályán elért maximum sebessége, adott pályán az átlag sebessége, adott pályán elért helyezése, adott pályán használt autó(k), befejezte-e a versenyt
 - Autók: azonosító, lóerő, csúcsebesség, ürtartalom, javítás alatt van.
 - Csapat: versenyzőinek listája, autók listája
 - Lehessen egy adott szezon pályái közül választani és lehessen lekérni a végeredmény adatait! Bármelyik pályát kiválasztva lehessen lekérni bármelyik versenyző, csapat vagy autó adatait!
 - Lehessen rendezetten megjeleníteni az adatokat. A felhasználó tudjon több mező alapján rendezni.
 - Hozzon létre egy grafikus felületet, amin ábrázolja a bemeneti adatokat.
 - Tudjon a program szűrni legalább két feltétel alapján! (csapat, kor)
 - A fent említett funkciókat jelenítse meg az Ön által választott események alapján (kattintásra, egér ráhúzásra, stb)
 - Extra, az Ön által kitalált funkciókat is építhet bele, ha a fentiek elkészültek!

Minta:

[illegible]

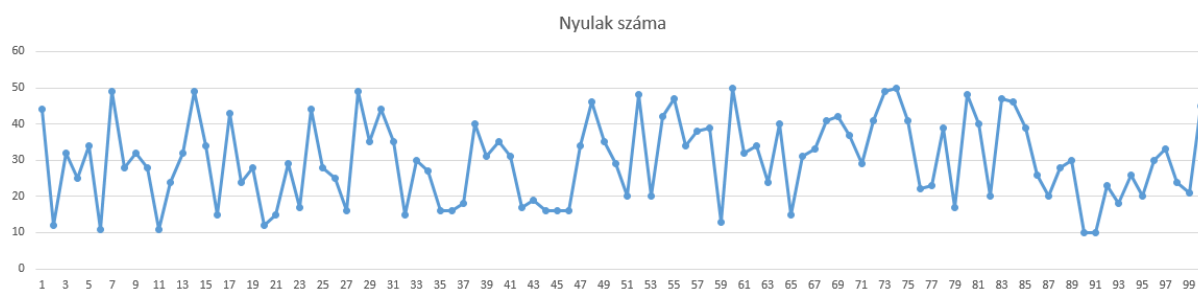
Nyulak és a mezők

5. **Projekt:** Írjon programot, amely a mezőn lévő nyulak életét szimulálja és vizsgálja.
- a. Adott egy vagy több szöveges állomány, amit Önnek kell létrehoznia a feladat leírás alapján! Az adatok beolvasása során a beolvasott adatokra adja meg a következő választ!
 - b. Írjon osztályokkal, metódusokkal programot, amit elvégzi a következő feladatok:
 - Lehessen lekérni a nyuszi aktuális helyzetét, korát!
 - Egy mezőre legyen feltöltve nyuszikkal. Ezt lehessen egy fájlból bekérni, de lehessen véletlenszerűen kigenerálni. A megjelenítő felület legyen egy kezdő állapot!
 - A szimuláció aktuális állapotát lehessen kimenteni külön fájlba, olyan szerkezettel, amit később felhasználhatók beolvasás során!
 - Minden beolvasás vagy kiíratás során legyen egy kommunikáló ablak a felhasználónak!
 - A szimulálás egy-egy lépés, amit minden nyuszin végre kell hajtani először.
 - Lépések:
 1. Születés: az üres mezőket végig járva, ha a környezetében pontosan három nyuszi van, akkor az adott területen új nyuszi születik, ezek a következő körben kerüljenek bele az adott adatszerkezetbe.
 2. Meghalás: ami bejárva a már meglévő nyusziakat, ha két vagy három nyuszi áll a szomszédságában akkor túléli és bekerül a következő listába, más különben a nyuszi a túlnépesedés miatt elpusztul.
 - A következő állapot listával írjuk felül a nyuszi listát.
 - *Figyeljünk oda, hogy a mező körkörös, azaz ha a nyúl bal szélén van akkor a bal szomszédja jobb szélén lévő érték.*
 - c. A szimulációhoz készítsen egy grafikus felületet. Készítsen a számok ábrázolásához grafikont is! Legyen lehetőségünk egy szimulált környezetet lépésről lépésre vizsgálni, illetve automatikusan is lehessen végig nézni egy szimulációt. Minden a két esetben egy grafikonon jelezze, hogy az adott állapotban épp mennyi a nyuszi születésének száma.

Minta kezdő állapotra 10x10 –es területen:

X									
		X			X	X	X		
		X				X			
		X		X	X				X
		X					X		
	X				X			X	
			X	X		X			X
X				X			X		
X				X		X		X	
					X				

Minta grafikonra 100. lépés után



Moziműsor

6. **projekt:** Írjon programot, amely a ceglédi Uránia mozi műsorait elemezi ki egy hét adatai alapján.
- a. Adott egy vagy több szöveges állomány, amit Önnek kell létrehoznia a feladat leírás alapján! Az adatok beolvasása során a beolvasott adatokra adja meg a következő választ!
 - b. Lehesse lekérni a Film magyar és angol címét, kategória listáját, hosszát, és a kiadott évét!
 - c. Lehesse lekérni a filmeket kategória alapján!
 - d. Kiíratásnál jelenítse meg a perc mellett, hogy az hány óra és hány perc.
 - e. Lehesse több (legalább két) terem közül választani! Minden teremnél legyen fix férőhely megadva! Legyen megadva a terem hossza és az adott sorban hány szék van!
 - f. A termékhez legyenek hozzárendelve filmek egymás után (szünet nélkül)
 - g. Lehesse időpontot foglalni a hétre. (csütörtök, péntek, ... kedd, szerda). A foglalásokat addig lehesse kezelni, amíg van szabad hely.
 - h. Lehesse lekérni egy adott nap filmjeit!
 - i. Lehesse rendezni a filmeket időpont, hossz, név stb. alapján!
 - j. Lehesse lekérni, hogy egy adott időpontban van-e éppen futó film, elkezdődött-e már a film.
 - k. Lehesse rákeresni megadott filmre, hogy játszó-e a héten!
 - l. Az adatokat fájlból olvassa be és fájlba írassa ki!
 - m. Hozzon létre egy grafikus felületet a feladat bemutatásához. Használjon táblázatot a megjelenítéshez. A megjelenítésben szabad kezet kap az ábrázoláshoz! Törekedjen arra, hogy minél több funkcióra minél több információ lekérésére és megjelenítésére lehesse használni!