

Junior Java backend fejlesztő házivizsga

Backend programozási feladatok

A belső vizsga projektje HackerRank-en keresztül, a megadott linken érhető el. A projektet importálja be, az 5 feladatot és egységtesztet valósítsa meg, majd az elkészült projektet kiexportálva, az elkészült zip fájlt töltsse vissza megoldásként.

Időtartam: 180 + 15 perc (letöltés; export/import; feltöltés; értelmezés többletidő)

Jó munkát!

I. feladat

Java programozási nyelv alapjai (feladat01 package)

Egyszerű titkosítást és dekódolást végző alkalmazást kell fejlesztenie. A titkosítás specifikációja:

- A bemeneti szövegen a következő átalakításokat kell elvégeznie a titkosító módszernek:
 - minden kis 'a' betűt cserélje le '1'-re
 - minden kis 'e' betűt cserélje le '2'-re
 - minden kis 't' betűt cserélje le '3'-ra
 - minden kis 's' betűt cserélje le '4'-re
 - minden kis 'k' betűt cserélje le '5'-re
 - minden szóközt cserélje le '6'-ra
 - A szöveg első karaktere kerüljön át a szöveg végére (nem másolás, hanem áthelyezés szükséges)
- Példa a generálásra:
 - titkosítandó szöveg = "Tesztszöveg ellenőrzése"
 - titkosítás eredménye → "24z34zöv2g62ll2nőrzé42T"

- a.) A példában használt adatokkal készült teszt (feladat01_Teszt package) nem fut le sikeresen. Javítsa a hibás vagy hiányos metódust!
- b.) Bővítse az alkalmazást és írja meg a dekódolást végző metódust!
- c.) Kérjen be titkosítatlan szöveget a konzolról majd írja ki a titkosított változatot! Hívja meg a dekódoló algoritmust és új sorban írja ki a visszakapott eredményt!
- d.) Írjon tesztet, ami a dekódoló metódust ellenőrzi!

II. feladat

Java objektumorientált programozás (feladat02 package)

Az alkalmazás egy botanikus kert növénygyűjteményét kezeli. A növényekről tárolt adatok: azonosító; megnevezés; latin neve; a botanikus kertbe kerülésének éve.

Noveny osztály rendelkezik egy metódussal, ami a lekérdezés pillanata és a bekerülés éve adatokból kiszámolja, az adott növény hány éve látogatható a botanikus kertben.

- a) A feladat02_Teszt package-ben lévő egységteszt nem fut le sikeresen. Javítsa a hibás vagy hiányos metódust!
- b) Hozza létre Noveny osztály leszármazottját, Fa osztályt, melynek saját adattagja a fa magasságát tartalmazza!
- c) Mindkét osztály rendelkezzen toString() metódus felülírással, amin keresztül az objektumok minden adata megjeleníthető legyen!
- d) A főprogramban hozzon létre egy 5 elemű tömböt, melyet töltsön fel tetszőleges értékű adatokkal! A példányok vegyesen, mindkét osztályt reprezentálják!
- e) A főprogramban egy külön metódus hívásával írja ki a képernyőre az objektumok minden adatát, a toString() metódusukat meghívva! Jelenjen meg az is, hány éve látogatható az adott növény. A sor végén kerüljön kijelzésre, hogy az adott példány általános növény vagy fa.
- f) Írjon további 1 tetszőleges tesztetet!

III. feladat

Java kollekciók (feladat03_JavaKollekciok package)

Egy autójavító szervíz árlistáját kezelő alkalmazást kell befejeznie. SzervizMunka osztály a szervíztevékenység megnevezését és a szervizeléshez szükséges munkaórák számát tartalmazza.

Az osztály arKepzes() metódusa kiszámolja az adott tevékenység munkadíját, a paraméterben megkapott óradíj felhasználásával. A program induláskor bekéri az aktuális óradíjat.

- a) A feladat03_Teszt package-ben lévő egységteszt nem fut le sikeresen. Javítsa a hibás vagy hiányos metódust!
- b) Az alkalmazás külön osztályban ciklikusan kérje be az adatokat (szervíztevékenység, órák száma) és képezzen objektumokat. A példányokat lista adatszerkezetben tárolja! A ciklusból kilépést -azaz nincs több rögzítendő tevékenység- az órák számának nulla értékkel megadásával jelezheti a felhasználó.
- c) Melyik szervíztevékenység tart a legrövidebb ideig? Írja ki a konzolra!
- d) Hozzon létre egy Map interfészt megvalósító adatszerkezetet és töltse fel az objektumok adataival. Az adatszerkezet kulcsa a szervíztevékenység megnevezése, míg értéke az arKepzes() metódus által számított munkadíj legyen! Listázza ki az adatokat!
- e) Bővítse az alkalmazást 1 új teszteset létrehozásával!

IV. feladat

Fájlkezelés Javaban (feladat04 package)

Egy papír írószer bolt szállítói rendelések nyilvántartását végző alkalmazást kell fejlesztenie. Az adatok a mellékelt SzallitoiRendelesek.csv fájlban találhatók. A fájl az adatokat - pontosvesszővel elválasztva- a következő sorrendben tárolja: szállítólevél azonosító; tétel megnevezés; mennyiség; összérték; sürgősségi szállítás (0 – nem, 1 igen).

A fájlt feldolgozó osztály beolvas metódusa egy numerikus egész értékkel tér vissza, ami a hibás sorok számát mutatja. Hibás sor az, ami kevesebb adatot tartalmaz az előírtnál, illetve a szállítólevél azonosítója rövidebb 5 karakternél. Az ilyen tételek nem példányosodnak, így adatszerkezetben sem tárolódnak. Az ellenőrzést az osztály ellenoriz metódusa végzi.

- a) feladat04_Teszt package-ben a beolvas metódusra vonatkozó egységteszt nem fut le sikeresen. Javítsa a hibás vagy hiányos metódust!
- b) Rendeles osztály konstruktorát bővítse úgy, hogy a létrejövő objektum megkapja a fájlból beolvasott értékeket!
- c) A főprogramban listázza ki az objektumokat, minden adatukkal!
- d) Adjon hozzá 2 új tételt a tároló adatszerkezethez, tetszőleges értékekkel! A megoldáshoz szükség szerint bővítse/módosítsa a meglévő osztályokat!
- e) Az így teljessé vált adatszerkezet minden adatát írja ki az UjSzallitoiRendelesek.csv fájlba! A fájl rendelkezzen egy fejléccel is! A megoldáshoz szükség szerint bővítse/módosítsa a meglévő osztályokat!
- f) Bővítse az alkalmazást 1 újabb tetszőleges tesztet létrehozásával!

V. feladat

Adatbáziskezelés Javaban (feladat05 package)

Egy színházi katalógus adatkezelését végző Swing ablakos alkalmazást kell befejeznie. Az adatok adatbázisból érkeznek. Az adatbázis (táblával és néhány adattal együtt) a projekt mappájában található szinhazAdatbazisLetrehoz.sql szkript futtatásával hozható létre. SzinhaziEloadas tábla attribútumai: előadás címe; a darab rendezője; bemutató dátuma; tervezett előadások száma.

- a. Futtassa a szkriptet! A kapcsolat felépítéséhez a mysql driver szintén a projekt mappájában található. Állítsa be a projekthez!
- b. Az alkalmazás indításkor építse fel az adatbázis kapcsolatot!
- c. feladat05_Teszt package-ben lévő egységtesztet nem fut le sikeresen. Javítsa a hibás vagy hiányos metódust!
- d. Helyezzen el a frame-n egy tetszőleges adatlistázásra alkalmas komponenst!
- e. Hozza létre a táblának megfelelő tároló osztályt! Az alkalmazás indulásakor automatikusan olvassa fel az adatbázist, tárolja az adatokat adatszerkezetben és jelenítse meg az előadások minden adatát az előzőekben létrehozott vezérlőn!
- f. Tegyen a frame-re egy „Törlés” feliratú gombot, melyre kattintva -amennyiben van kijelölt elem- dialógusablakban kérdezzen rá a törlési szándékra. A felhasználó igenlő válasza esetén törölje az adatot a vezérlőről, az adatszerkezetből és az adatbázisból egyaránt!
- g. Bővítse az alkalmazást 1 újabb tetszőleges tesztet létrehozásával!