

Junior Java backend fejlesztő házivizsga

Backend programozási feladatok

A belső vizsga projektje pCloud-on, a megadott linken érhető el. A projektet importálja be, az 5 feladatot és egységtesztet valósítsa meg, majd az elkészült projektet (vagy a kiexportált projekt zip fájlját) tölts fel saját GitHub fiókjába. A projekthez a tanult módon hívjon meg kollaborátorként (felhasználói azonosítóm: BBotlik), ezen keresztül fogom letölteni és javítani a megoldást.

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

Jó munkát!

I. feladat

Java programozási nyelv alapjai (feladat01 package)

Jelszavak erősségét ellenőrző algoritmust kell megvalósítani. A jelszavak tömbben vannak tárolva, a programnak kell megállapítania, hogy a követelményeknek hány jelszó felel meg. Az erős jelszó jellemzői a specifikáció szerint:

- pontosan 10 karakter hosszú
- az első karakter nagybetű
- másodiktól nyolcadik karakterig kisbetű
- az utolsó két karakter számjegy

Az erősséget ellenőrző metódus igaz/hamis eredményt szolgáltat a jelszó erősségével kapcsolatban.

- a.) A feladat01_Teszt package-ben lévő egységteszt nem fut le sikeresen. Javítsa a tesztelt -hibás vagy hiányos- metódust!
- b.) A főprogramban tárolt tömbből állapítsa meg és írassa ki az alkalmazással, hogy hány érvényes jelszót tartalmaz. Ehhez adja át a tömböt az "ervenyes" metódusnak, ahol egy ciklussal menjen végig a tömbön és minden elemet vizsgáltsa meg az "ellenorzes" metódussal. Az eredménynek megfelelően írja ki egyesével a tömbelemeket és mögé az "érvényes" vagy "érvénytelen" megjegyzést. Közben számolja össze az érvényes jelszavakat, ez legyen a visszatérési érték a főprogramba.
- c.) Írjon tesztet az "ervenyes" metódus működésére vonatkozólag!

II. feladat

Java objektumorientált programozás (feladat02 package)

Egy édességeket forgalmazó cég nyilvántartását kell szoftveresen megoldani. Minden édességről tudjuk a következőket: megnevezés, egységár, darabszám
Edesseg osztály rendelkezik egy metódussal, ami visszaadja a termék aktuális készletértékét.

- a) A feladat02_Teszt package-ben lévő egységteszt nem fut le sikeresen. Javítsa a tesztelt -hibás vagy hiányos- metódust!
- b) Készítse el Edesseg osztály leszármazottját, Csokolade osztályt, melynek saját adattagja a kakaótartalom legyen! Edesseg osztály másik leszármazottja pedig legyen Cukorka osztály, saját adattagja egy logikai változó (töltött/nem töltött). Sem Csokolade, sem Cukorka osztály nem származtatható tovább!
- c) Mindhárom osztály rendelkezzen toString() metódus felülírással, amin keresztül az objektumok minden adata megjeleníthető legyen!
- d) A főprogramban hozzon létre egy 3 elemű tömböt, melyet töltsön fel tetszőleges értékű adatokkal! A példányok vegyesen, mindhárom 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, hogy az egyes termékeknek mennyi a készletértéke. A sor végén kerüljön kijelzésre az adott példány típusa (Edesseg,Csokolade,Cukorka).
- f) Írjon tesztetet Csokolade vagy Cukorka osztály toString() metódusára vonatkozóan!

III. feladat

Java kollekciók (feladat03_JavaKollekciok package)

Egy társasház parkolójába csak a lakók autói állhatnak be. Ehhez egy alkalmazást használnak, ami naprakészen tárolja a behajtható autók rendszámait. A rendszámokat a főprogramban olyan adatszerkezetben tárolja, amely biztosítja, hogy nem lehet rendszám duplikáció! A program ezen verziója módosítást és törlést egyelőre nem tud végezni, csak adatfelvitelt.

A főprogram behajthat() metódusa ellenőrzi, hogy a paraméterben átvett rendszám tárolva van-e a rendszerben és igaz/hamis eredményt szolgáltat a behajtásra vonatkozóan.

- a) A feladat03_Teszt package-ben lévő egységteszt nem fut le sikeresen. Javítsa a tesztelt -hibás vagy hiányos- metódust!
- b) Valósítsa meg a főprogramban található adatfelvitel menüpont által hívott metódus algoritmusát, ahol ciklikusan kérje be és töltse be az adatszerkezetbe a rendszámokat, amíg a felhasználó "VÉGE" adatot nem ad meg rendszámként. Ekkor térjen vissza a főprogram menüjébe, visszaadva az újonnan sikeresen felvitt rendszámok darabszámát, ami ki is íródik a képernyőre.
- c) Kilépéskor írjon ki a képernyőre statisztikát: hány rendszám került tárolásra, illetve hány autó hajthatott be és hány nem. Ehhez a megfelelő helyeken vezesse be a statisztikai változókat és a hozzá tartozó algoritmust.
- d) Bővítse az alkalmazást 1 új teszt eset 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, vagy a szállítólevél azonosítója nem felel meg annak a szabálynak, miszerint az azonosító 'S' betűvel kezdődik és pontosan 6 karakter hosszú. 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 tesztelt -hibás vagy hiányos- metódust!
- b) Rendes osztály meglévő konstruktorát bővítse a paraméterezésnek megfelelően! A konstruktort a fájl beolvasását végző alprogram hívja meg.
- c) A főprogramban listázza ki az objektumokat, minden adatukkal, fordított sorrendben (az utolsó elemtől az első elemig)
- d) Adjon hozzá 2 új tételt a tároló adatszerkezethez, tetszőleges értékekkel! A megoldáshoz szükség szerint bővítse/módosítsa a meglévő osztályokat! Az új tételek is kerüljenek kiírásra ugyanabba a fájlba, hozzáfűzéssel!
- e) 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 egységteszt nem fut le sikeresen. Javítsa a tesztelt -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 „Kalkuláció” feliratú gombot. A gombra kattintva -ha van kiválasztott adat a megjelenítő vezérlőn- ugorjon fel egy OptionPane és mutasson egy kalkulációt, mennyi lehet a maximum bevétel, előadásonként 2 millió bevétellel számolva. Ha nincs kiválasztott elem, küldjön figyelmeztetést!
- g. Bővítse az alkalmazást 1 újabb tetszőleges tesztet létrehozásával!