

Az alábbi feladatok típusait egy-egy osztály segítségével valósítsa meg. Egy osztály szolgáltatásainak (metódusainak) bemutatásához olyan főprogramot kell készíteni, amelyik egy menü segítségével teszi lehetővé a metódusok tetszőleges sorrendben történő kipróbálását. A főprogram ezért példányosítson legalább egy objektumot, amelyre a menüpontok közvetítésével lehet meghívni az egyes metódusokat. Természetesen szükség lesz minden tevékenység után az objektum állapotának kiírására vagy egy az objektum állapotát kiíró külön menüpontra. Azoknál az osztályoknál, ahol vannak olyan metódusok (esetleg barát függvények), amelyek több objektum közötti műveleteket valósítanak meg, a főprogram több objektum létrehozására és azok állapotának kiírására adjon lehetőséget.

1. Készítsen egy sor típust! Alkalmazzon osztályt! A sort dinamikusan lefoglalt tömb segítségével ábrázolja! Törekedjen a sor reprezentálásánál a tömb minél jobb kihasználására! Implementálja a szokásos műveleteket, alkalmazzon kivételkezelést és bontsa modulokra a programját! A teszt környezet adjon lehetőséget egy sor létrehozására, feltöltésére, a sor műveleteinek kipróbálására! Tegye lehetővé két sor típusú változó közötti értékadást!
2. Valósítsa meg a felsőháromszög mátrixtípust (a mátrixok a főátlójuk alatt csak nullát tartalmaznak)! Ilyenkor elegendő csak a főátló és afeletti elemeket reprezentálni egy sorozatban, amelyet egy dinamikus helyfoglalású tömbben helyezünk el. Alkalmazzon operátor túlterhelést és kivételkezelést! Implementáljuk a mátrix i -edik sorának j -edik elemét visszaadó műveletet, valamint a hatékony összeadás és szorzás műveleteket! Tegye lehetővé két felsőháromszög-mátrix típusú változó közötti értékadást!
3. Valósítsa meg a nagyon nagyszámok típusát! Ábrázoljuk a számokat számjegyeik sorozatával, amelyet egy dinamikus helyfoglalású tömbben helyezünk el, és implementáljuk a hatékony összeadás és a szorzás műveleteit! Alkalmazzon operátor túlterhelést és kivételkezelést! Tegye lehetővé két nagyszám típusú változó közötti értékadást!
4. Valósítsa meg a polinomok típusát! Ábrázoljuk a polinomokat együtthatóik sorozatával, amelyet egy dinamikus helyfoglalású tömbben helyezünk el, és implementáljuk a hatékony összeadás és a szorzás műveleteit! Alkalmazzon operátor túlterhelést és kivételkezelést! Tegye lehetővé két polinom típusú változó közötti értékadást!
5. Készítsen egy halmaz típust! Alkalmazzon osztályt! A halmazt dinamikusan lefoglalt tömb segítségével ábrázolja! Implementálja a szokásos műveleteket (elem betevése, kivétele, üres-e a halmaz, egy elem benne van-e a halmazban, kiírás és beolvasás operátor $<<$ és operátor $>>$ segítségével), alkalmazzon kivételkezelést és bontsa modulokra a programját! A teszt környezet adjon lehetőséget két halmaz létrehozására, feltöltésére, a halmaz műveleteinek kipróbálására, és hívjon meg egy olyan barát-operátort is, amely kiszámítja két halmaz unióját! Törekedjen az unióképzés műveletigényének minimalizálására, a dokumentációban mutasson rá a saját megoldásának műveletigényére! Tegye lehetővé két halmaz típusú változó közötti értékadást!
6. Készítsen egy zsák típust! Alkalmazzon osztályt! A zsákokat dinamikusan lefoglalt tömb segítségével ábrázolja! Implementálja a szokásos műveleteket (elem betevése, kivétele, üres-e a zsák, egy elem hányszor van a zsákban, kiírás és beolvasás operátor $<<$ és operátor $>>$ segítségével), alkalmazzon kivételkezelést és bontsa modulokra a programját! A teszt környezet adjon lehetőséget két zsák létrehozására, feltöltésére, a zsák műveleteinek kipróbálására, és hívjon meg egy olyan barát-

operátort is, amely kiszámítja két zsák unióját (a közös elemek előfordulása összegződik)! Törekedjen az unióképzés műveletigényének minimalizálására, a dokumentációban mutasson rá a saját megoldásának műveletigényére! Tegye lehetővé két zsák típusú változó közötti értékadást!

7. Készítsen egy halmaz típust! Alkalmazzon osztályt! A halmazt dinamikusán lefoglalt tömb segítségével ábrázolja! Implementálja a szokásos műveleteket (elem betevése, kivétele, üres-e a halmaz, egy elem benne van-e a halmazban, kiírás és beolvasás operátor<< és operátor>> segítségével), alkalmazzon kivételkezelést és bontsa modulokra a programját! A teszt környezet adjon lehetőséget két halmaz létrehozására, feltöltésére, a halmaz műveleteinek kipróbálására, és hívjon meg egy olyan barát-operátort is, amely kiszámítja két halmaz metszetét! Törekedjen a metszetképzés műveletigényének minimalizálására, a dokumentációban mutasson rá a saját megoldásának műveletigényére! Tegye lehetővé két halmaz típusú változó közötti értékadást!
8. Készítsen egy zsák típust! Alkalmazzon osztályt! A zsákokat dinamikusán lefoglalt tömb segítségével ábrázolja! ! Implementálja a szokásos műveleteket (elem betevése, kivétele, üres-e a zsák, egy elem hányszor van a zsákban, kiírás és beolvasás operátor<< és operátor>> segítségével), alkalmazzon kivételkezelést és bontsa modulokra a programját! A teszt környezet adjon lehetőséget két zsák létrehozására, feltöltésére, a zsák műveleteinek kipróbálására, és hívjon meg egy olyan barát-operátort is, amely kiszámítja két zsák metszetét (a közös elemek a kisebb előfordulási számmal maradnak meg)! Törekedjen a metszetképzés műveletigényének minimalizálására, a dokumentációban mutasson rá a saját megoldásának műveletigényére! Tegye lehetővé két zsák típusú változó közötti értékadást!
9. Készítsen egy halmaz típust! Alkalmazzon osztályt! A halmazt dinamikusán lefoglalt tömb segítségével ábrázolja! Implementálja a szokásos műveleteket (elem betevése, kivétele, üres-e a halmaz, egy elem benne van-e a halmazban, kiírás és beolvasás operátor<< és operátor>> segítségével), alkalmazzon kivételkezelést és bontsa modulokra a programját! A teszt környezet adjon lehetőséget két halmaz létrehozására, feltöltésére, a halmaz műveleteinek kipróbálására, és hívjon meg egy olyan barát-operátort is, amely kiszámítja két halmaz különbségét! Törekedjen a különbségképzés műveletigényének minimalizálására, a dokumentációban mutasson rá a saját megoldásának műveletigényére! Tegye lehetővé két halmaz típusú változó közötti értékadást!
10. Készítsen egy zsák típust! Alkalmazzon osztályt! A zsákokat dinamikusán lefoglalt tömb segítségével ábrázolja! ! Implementálja a szokásos műveleteket (elem betevése, kivétele, üres-e a zsák, egy elem hányszor van a zsákban, kiírás és beolvasás operátor<< és operátor>> segítségével), alkalmazzon kivételkezelést és bontsa modulokra a programját! A teszt környezet adjon lehetőséget két zsák létrehozására, feltöltésére, a zsák műveleteinek kipróbálására, és hívjon meg egy olyan barát-operátort is, amely kiszámítja két zsák különbségét (a közös elemek csak akkor maradnak meg, ha a kivonandó zsákbeli előfordulási számuk a nagyobb, és ekkor az előfordulási számuk különbségével)! Törekedjen a különbségképzés műveletigényének minimalizálására, a dokumentációban mutasson rá a saját megoldásának műveletigényére! Tegye lehetővé két zsák típusú változó közötti értékadást!