

11. Objektum-orientált programozás

1. Írjunk programot, amely két valós adatnak kiszámítja a mértani közepét! (*\MINTA_01*)
2. Írjunk programot, amely **rekord** használatával számítja ki két valós szám mértani közepét! (*\MINTA_02*)
3. Írjunk objektum-orientált programot, amely két valós szám mértani közepét számítja ki modul használatával! (*\MINTA_03*)
4. Módosítsuk a *\MINTA_03* könyvtárban lévő programot, hogy a főprogram dinamikus objektumpéldányt használjon! (*\MINTA_04*)
5. Módosítsuk a *\MINTA_04* könyvtárban lévő programot, hogy az objektumnak adatmezői dinamikusak legyenek, főprogram dinamikus objektumpéldányt használjon! (*\MINTA_05*)
6. Módosítsuk a *\MINTA_05* könyvtárban tárolt programot, hogy a *badat* objektumot az *adat* objektumból származtassuk és a *csokken* metódusa az *x* és *y* adatmezőit felére csökkentse! (*\MINTA_06*)
7. Módosítsuk a *\MINTA_06* könyvtárban tárolt programot, úgy hogy a *badat* objektumnak legyen egy *Szamol* metódusa, amely az *ös* objektum azonos nevű metódusát felülbírálja, a feladata számtani közép számítása! (*\MINTA_07*)
8. Módosítsuk a *\MINTA_07* könyvtárban tárolt programot, úgy hogy az *adat* objektum *vegrehajt* metódusa hívja a saját *szamol* metódusát és ezt a metódust hívja a főprogramban a két objektumpéldány! Mit tapasztalunk? (*\MINTA_08*)
9. Módosítsuk a *\MINTA_08* könyvtárban lévő programot, úgy hogy az *adat* objektum *szamol* metódusa és a *badat* objektum *szamol* metódusa legyen **virtual**. Mit tapasztalunk? (*\MINTA_09*)
10. Írjunk objektum-orientált programot, amely komplex számokkal elvégzi a négy alpműveletet! (*KOMPLEX\kompl_o*)
11. Írjunk objektum-orientált, menüvezérelt programot, amely komplex számokkal elvégzi a négy alpműveletet! (*KOMPLEX\komplo_m*)

12. Módosítsuk a *komplo_m* programot, úgy hogy az objektumokat helyezzük *komp_l_u* könyvtárba és próbáljuk ki a *komp_l_pr* főprogramban! (*KOMPLEX\komp_l_u, komp_l_pr*)
13. Írjunk objektum-orientált programot tanulók nevét névsorban rendezi, a születési dátumukat pedig csökkenő sorrendbe rendezi! A főprogramban statikus objektum-példányt használjunk! (*RENDEZ\rendezp1*)
14. Írjunk objektum-orientált programot tanulók nevét névsorban rendezi, a születési dátumukat pedig csökkenő sorrendbe rendezi! Az objektumban dinamikus adatmezőt és a főprogramban dinamikus objektum-példányt használjunk! (*RENDEZ\rendezp2*)
15. Írjunk objektum-orientált programot, amely 20 valós adatoknak összegét és átlagát képezi, megkeresi a legnagyobb és a legkisebb elemét! Az objektumban statikus adatmezőket és a főprogramban statikus objektum-példányt használjunk! (*VEKTOR\vektor_s*)
16. Írjunk objektum-orientált programot, amely 20 valós adatoknak összegét és átlagát képezi, megkeresi a legnagyobb és a legkisebb elemét! Az objektumban dinamikus adatmezőt és a főprogramban dinamikus objektum-példányt használjunk! (*VEKTOR\vektor_p*)
17. Írjunk objektum-orientált programot, amely beolvas egy mondatot, felbontja szavakra, kiírja a szó hosszát, a magánhangzók és a mássalhangzók számát táblázatosan. (*MONDAT\mondat_o*)
18. Írjunk objektum-orientált programot, amely a kör sugara alapján kiszámítja a kör területét és területét! A főprogramban statikus objektumpéldányt használjunk! (*KOR\oop2*)
19. Írjunk objektum-orientált programot, amely a kör sugara alapján kiszámítja a kör területét és területét! A főprogramban dinamikus objektumpéldányt használjunk! (*KOR\oop2d*)
20. Írjunk objektum-orientált programot, amely a kör sugara alapján kiszámítja a kör területét és területét! Az objektumban dinamikus adatmezőket és a főprogramban dinamikus objektumpéldányt használjunk! (*KOR\oop2dd*)