A csoport

1. Egy vadásztársaság több vadászatot rendez, és a vadászatokon több vadász indul. Ugyanaz a vadász több vadászaton is részt vehet. A vadászatoknak van egy azonosítója, és ismert a helyszíne. A vadászokról tudjuk, hogy az egyes vadászatokon melyek voltak a trófeáik. Egy trófea leírja, hogy egy adott vadászaton elejtett vadnak mi a fajtája és súlya (kg-ban). A vadak fajtája lehet szarvas, vaddisznó, vagy medve. Egy trófea értéke függ az elejtett vad súlyától, valamint egy veszélyességi tényezőtől, amivel a súlyt meg kell szorozni. Ez a tényező a szarvasoknál 1, vaddisznóknál 1.5, medvéknél 3.
2. Adja meg a fenti probléma szerkezetét leíró osztálydiagrammot! Tüntesse fel az osztályok adattagjait, ábrázolja az objektumok közötti kapcsolatokat, helyezzen el szerepneveket, amelyekkel egy objektum hivatkozni tud egy vele kapcsolatban álló objektumra vagy objektumokra! (15 pont)
3. Definiálja egy vad értékét kiszámoló metódust! Alkalmazzon tervezési mintákat is, ha tud, és nevezze is meg azokat! (10 pont)
4. Tervezzen két további metódust és helyezze ezeket a megfelelő osztályokba! Az egyik számolja ki, hogy mennyi volt egy vadász trófeáinak összértéke egy adott azonosítójú vadászaton. A másik keresse meg egy vadászat győztesét (a trófeái értékének összesítése szerinti legjobb vadászt). A metódusok algoritmusát tanult programozási tételből származtassa: nevezze meg a tételt, adja meg a megfeleltetési táblázatát a konkrét tulajdonságokkal és a felsorolóval. Az algoritmust pszeudo kóddal is leírhatja! (15 pont)
5. Tekintsen egy korlátos halmaz típusú objektumot! Ez megadott számú *Item* típusú elem tárolására alkalmas úgy, hogy egy *max* méretű (*max*≥2) *t* tömbben tárolja az elemeket, amelyek folyamatosan töltik fel a tömböt egy *ind* indexig bezárólag (*ind*≤*max*). A megengedett műveletei:

*add(e)* – beteszi az *e* elemet a halmazba (ha még nincs benne és a halmaz nincs tele),

*remove(e)* – kiveszi az *e* elemet a halmazból (ha benne van a halmazban),

*in(e)* – eldönti, hogy az *e* elem benne van-e a halmazban,

*empty()* – eldönti, hogy a halmaz üres-e,

*full()* – eldönti, hogy a halmaz tele van-e.

Adja meg a halmaz osztályát és rajzolja fel az állapotgépét! (20 pont)

B csoport

1. Egy országos horgászverseny keretében több versenyt rendeznek, és ezeken a versenyeken több horgász indul, ugyanaz a horgász akár több versenyen is részt vehet. A versenyek egyedi azonosítóval rendelkeznek, és ismert a helyszínük. A horgászokról ismert, hogy az egyes versenyeken milyen fogásaik voltak. Egy fogás leírja, hogy egy adott versenyen kifogott halnak mi fajtája és a mérete (súly kg-ban). A halak fajtája lehet ponty, keszeg, vagy harcsa. Egy fogás értéke függ a kifogott hal súlyától, valamint egy módosító tényezőtől, amivel a súlyt kell megszorozni. Ez a módosító tényező a pontyoknál 1.5, a keszegeknél 4.5, a harcsáknál 0.2.
2. Adja meg a fenti probléma szerkezetét leíró osztálydiagrammot! Tüntesse fel az osztályok adattagjait, ábrázolja az objektumok közötti kapcsolatokat, helyezzen el szerepneveket, amelyekkel egy objektum hivatkozni tud egy vele kapcsolatban álló objektumra vagy objektumokra! (15 pont)
3. Definiálja egy fogás értékét kiszámoló metódust! Alkalmazzon tervezési mintákat is, ha tud, és nevezze is meg azokat! (10 pont)
4. Tervezzen két további metódust és helyezze ezeket a megfelelő osztályokba! Az egyik számolja ki, hogy mennyi volt egy horgász fogásainak összértéke egy adott azonosítójú versenyen. A másik keresse meg egy horgászverseny győztesét (a fogásai összértéke szerinti legjobb horgászt). A metódusok algoritmusát tanult programozási tételből származtassa: nevezze meg a tételt, adja meg a megfeleltetési táblázatát a konkrét tulajdonságokkal és a felsorolóval. Az algoritmust pszeudo kóddal is leírhatja! (15 pont)
5. Tekintsen egy korlátos zsák típusú objektumot! Ez megadott számú különböző *Item* típusú elem tárolására alkalmas úgy, hogy egy *max* méretű (*max*≥2) *t* tömbben tárolja az elemeket a zsákban való előfordulásukkal együtt, és ezen elem-darabszám párok folyamatosan töltik fel a tömböt egy *ind* indexig bezárólag (*ind*≤*max*). Egy *max* féle elemet tároló zsák már tovább nem bővíthető olyan elemmel, amely még nem szerepel a zsákban. A zsák megengedett műveletei:

*putin(e)* – beteszi az *e* elemet a zsákba (ha *e* már benne vagy ha a zsák bővíthető),

*takeout(e)* – kiveszi az *e* elemet a zsákból (ha benne van a zsákban),

*in(e)* – eldönti, hogy az *e* elem benne van-e a zsákban,

*empty()* – eldönti, hogy a zsák üres-e,

*full()* – eldönti, hogy a zsák bővíthető-e újabb elemmel.

Adja meg a halmaz osztályát és rajzolja fel az állapotgépét! (20 pont)