C csoport

1. Egy vidámpark több céllövöldét is üzemeltet, ahol a vendégek szerencsét próbálhatnak. Egy céllövöldében több vendég is lőhet akár többször is. A céllövöldék egyedi azonosítóval rendelkeznek, és ismert a helyszínük. A vendégek is kapnak egy egyedi azonosítót, valamint ismert, hogy az egyes céllövöldékben melyek voltak a nyereményeik. Egy nyeremény leírja, hogy egy adott azonosítójú céllövöldében elnyert ajándéknak mi a típusa és a mérete (S, M, L, XL). Az ajándék típusa lehet plüss állat, műanyag figura, vagy labda. Az S méret 1 pontot, az M 2 pontot, az L 3 pontot, az XL 4 pontot ér. Az ajándék pontszáma plüss állat esetén 3 pont, műanyag figura esetén 2 pont, labda esetén 1 pont. Egy nyeremény értékét az ajándék típusából és méretéből származó pontok szorzata adja.
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 nyeremény 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 vendég nyereményeinek összértéke egy adott céllövöldében. A másik keresse meg egy céllövölde (nyereményei értékének összesítése szerinti) legeredményesebb vendégét. 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 elsőbbségi sor típusú objektumot! Ez megadott számú *Item* típusú elem tárolására alkalmas úgy, hogy egy *max* méretű (*max*≥2) *t* tömbben tárolja az elemeket azok értékének csökkenően rendezett sorrendjében úgy, hogy az elemek folyamatosan töltik fel a tömböt egy *ind* indexig bezárólag (*ind*≤*max*). Az elsőbbségi sor megengedett műveletei:

*add(e)* – beteszi az *e* elemet a sorba (ha a sor még nincs tele),

*remMin()* – kiveszi a sor egyik legkisebbelemet (ha a sor nem üres),

*min()* – visszaadja a sor egyik legkisebb elemét (ha a sor nem üres),

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

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

Adja meg az elsőbbségi sor osztályát és rajzolja fel az állapotgépét! (20 pont)

D csoport

1. Egy országos rally autóverseny több futamból áll, a futamokon több csapat indul. Ugyanaz a csapat több futamon is részt vehet. A futamok egyedi azonosítóval rendelkeznek, és ismert a helyszínük. A csapatok több kategóriában versenyezhetnek. A csapatok is kapnak egy egyedi azonosítót, és ismert, hogy az egyes futamokon milyen eredményt értek el. Egy eredmény leírja, hogy egy adott azonosítójú futamon az melyik kategóriában hányadik helyezés volt. A kategória lehet sportautó, teherautó, motor. Egy eredmény értéke függ az elért helyezéstől, valamint egy módosító tényezőtől, amivel a helyezést kell megszorozni. Ez a módosító tényező a motoroknál 1, sportautóknál 1.5, teherautóknál 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 eredmény 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 csapat összeredménye egy adott futamon. A másik keresse meg egy futam legeredményesebb csapatát (akinek az összeredménye legkisebb). 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 sor típusú objektumot! Ez megadott számú *Item* típusú elem tárolására alkalmas úgy, hogy egy *max* méretű (*max*≥2) *0*-tól kezdődően indexelt *t* tömbben tárolja az elemeket a beérkezésük sorrendjében. A *count* számú tárolt elem (*count*≤*max*) a tömbben az *out*-adik indexű helyétől kezdődően az *in‒1*-edik indexű helyig helyezkedik el (*out, in*∊[*0*..*max*]), de ciklikusan, azaz ha *count*>*0* és *out*>*in*, akkor a sor első szakaszát az *out*..*max‒1* indexű helyeken, a sor második szakaszát a *0*..*in‒1* indexű helyeken találjuk. A sor megengedett műveletei:

*push(e)* – beteszi az *e* elemet a sor végére (ha a sor még nincs tele),

*pop()* – kiveszi a sor elsőelemét (ha a sor nem üres),

*top()* – visszaadja a sor első elemét (ha a sor nem üres),

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

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

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