

Programozás II. 1. ZH

SZTE Szoftverfejlesztés Tanszék
2022. ősz

Technikai ismertető

- A programot C++ nyelven kell megírni.
- A megoldást a *Bíró* fogja kiértékelni.
 - A Feladat beadása felületen a Feltöltés gomb megnyomása után ki kell várni, amíg lefut a kiértékelés. **Kiértékelés közben nem szabad az oldalt frissíteni vagy a Feltöltés gombot újból megnyomni** különben feltöltési lehetőség veszik el!
- Feltöltés után a *Bíró* a programot g++ fordítóval és a
-std=c++1y -static -O2 -DTEST_BIR0=1
paraméterezéssel fordítja és különböző tesztesetekre futtatja.
- A program működése akkor helyes, ha a tesztesetek futása nem tart tovább 5 másodpercnél és hiba nélkül (0 hibakóddal) fejeződik be, valamint a program működése a feladatkiírásnak megfelelő.
- A *Bíró* által a `riport.txt`-ben visszaadott lehetséges hibakódok:
 - Futási hiba 6: Memória- vagy időkorlát túllépés.
 - Futási hiba 8: Lebegőpontos hiba, például nullával való osztás.
 - Futási hiba 11: Memória-hozzáférési probléma, pl. tömb-túindexelés, null pointer használat.
- A `riport.txt` és a fordítási log fájlok megtekinthetők az alábbi módon:
 1. Az Eredmények megtekintése felületen a vizsgálandó próba új lapon való megnyitása
 2. A kapott url formátuma:
`https://biro.inf.u-szeged.hu/Hallg/IBL302g-1/1/hXXXXXX/4/riport.txt`
 3. Az url-ből visszatörölve a 4-esig (`riport.txt` törlése) megkaphatók a 4-es próbálkozás adatai
- A programot 20 alkalommal lehet benyújtani, a megadott határidőig.
- A programban szerepelhet `main` függvény, amely a pontszámításkor nem lesz figyelembe véve. Azonban ha fordítási hibát okozó kód van benne az egész feladatsor 0 pontos lesz.

Általános követelmények, tudnivalók

- Csak a leírásban szereplő osztályokat, metódusokat és adattagokat kell megvalósítani, egyéb dolgokért nem jár plusz pont.
- Minden metódus, amelyik nem változtatja meg az objektumot, legyen konstans! Ha a paramétert nem változtatja a metódus, akkor a paraméter legyen konstans!
- string összehasonlításoknál az egyezés a pontos egyezést jelenti, azaz ha kis-nagy betűben térnek el, akkor már nem tekinthetők egyenlőnek (pl. a "piros" != "Piros")
- A leírásokban bemutat példákban a string-ek köré rakott idézőjelek nem részei az elvárt kimenetnek, azok csak a string határait jelölik. Például ha az szerepel, hogy a példa bemenetre az elvárt kimenet az, hogy "3 alma", akkor az elvárt kimenet idézőjelek nélkül az 3 alma, de a szóköz szükséges!
 - A tesztesetekben nem lesz ékezetes szöveg kiírása.
- Az elvárt kimeneteknek karakterről karakterre olyan formátumúnak kell lennie, ami a feladatban le van írva (szóközöket és sortöréseket is beleértve).

Hamarosan Húsvét!

Ügyelj rá, hogy minden olyan metódus konstans legyen, ami nem módosít az adattagok értékein! Ha egy metódus nem változtat a paraméterén, akkor az legyen konstans!

1. feladat: Nyuszi (11 pont)

Készítsd el a `Nyuszi` osztályt, amely egy nyuszit fog reprezentálni. **(1+0 pont)**

A nyuszi adattagjait az alábbi táblázat mutatja be:

Adattag neve	Típusa	Jelentése	Getter neve	Setter neve
nev	string	a nyuszi neve	get_nev	set_nev
sebesseg	unsigned	a nyuszi sebessége	get_sebesseg	set_sebesseg

Az adattagok csak az osztályból legyenek elérhetőek, de készíts hozzájuk getter és setter metódusokat a fent látható táblázat szerint. **(1+1 pont)**

Készíts az osztályhoz egy default konstruktort, ami a nevet *Muszira*, és a sebességet 90-re állítja. **(0+1 pont)**

Készíts az osztályhoz egy paraméteres konstruktort is, amely a 2 adattag értékét várja paraméterben és inicializálja az adattagokat azok alapján. A konstruktor paramétereinek sorrendje: *nev*, *sebesseg* **(0+2 pont)**

Definiáld felül az osztályban a *prefix* és *postfix* `++` operátort, amelyek a nyuszi sebességét megnövelik 1-gyel. Figyelj oda a postfix működés helyes megvalósítására! **(2+3 pont)**

2. feladat: A nyuszik répát esznek (10 pont)

Készítsd el a **Repa** osztályt, amely egy répát reprezentál, amelyet a nyuszik elfogyasztanak.

A répa adattagjait az alábbi táblázat mutatja be:

Adattag neve	Típusa	Jelentése	Getter neve	Setter
meret	float	a répa nagysága cm-ben	get_meret	-
nyuszik	Nyuszi[5]	a répát fogyasztó nyuszik	get_nyuszik	-
nyusziszam	unsigned	jelenleg hány nyuszi fogyasztja a répát	get_nyusziszam	-

Az adattagok csak az osztályból és a gyerekosztályokból legyenek elérhetőek, de készíts hozzájuk getter metódusokat! **(0+1 pont)**

Az osztálynak legyen egy 1 paraméteres konstruktora, amely a répa méretét várja és az alapján inicializálja az adattagot. Kezdetben a répát 0 nyuszi fogyasztja. Minden répa legalább 10 cm nagyságú. Ha ennél kisebb értéket kapunk, akkor az adattag értékét állítsuk 10-re. **(0+2 pont)**

Definiáld felül a **Repa** osztályban a `+=` operátort, amely segítségével nyuszit lehet hozzáadni a répához. Hozzáadáskor a paraméterben kapott nyuszi a tömb legelső üres pozíciójára kerüljön!

Köztudott, hogy egy adott répát fogyasztó nyuszik nevének különbözőnek kell lennie. Amennyiben a tömb tartalmaz már olyan nyuszit, amelynek neve megegyezik a paraméterben érkező nyuszi nevével, akkor a metódus ne csináljon semmit.

Amennyiben a tömb megtelt, akkor az objektum ne változzon, de írjuk ki a standard out-putra az alábbi szöveget (idézőjelek nélkül), sortöréssel a végén:

"A repa mar fogyasztas alatt all."

A metódus adjon vissza referenciát az aktuális (módosított) objektumra! **(0+4 pont)**

Lehessen az objektumot stringgé konvertálni. A konverzió eredménye a leggyorsabb nyuszi neve legyen! Amennyiben a répát egy nyuszi sem fogyasztja, akkor üres stringet adjunk vissza! **(1+2 pont)**