

Programozás I. 3. zh

SZTE Szoftverfejlesztés Tanszék

2024. tavasz

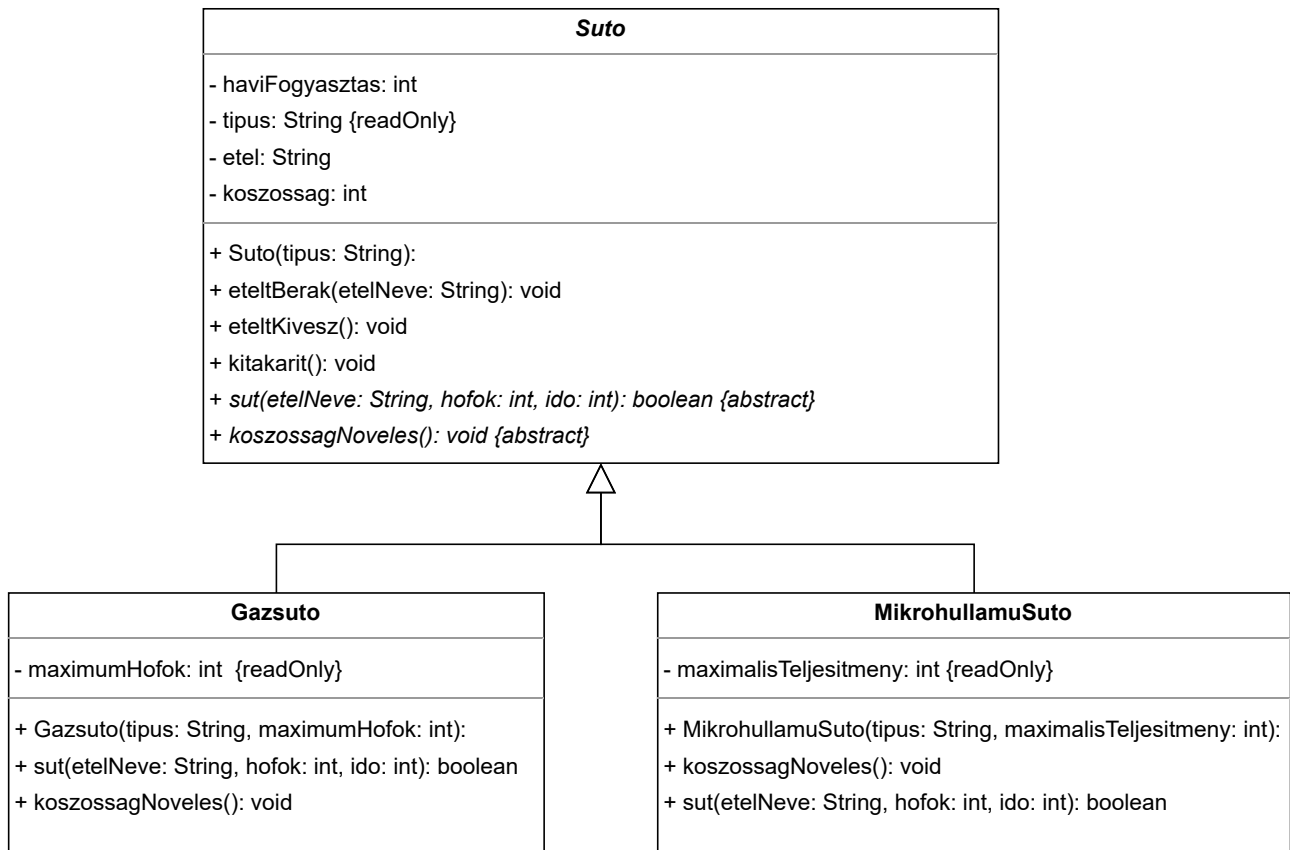
Általános követelmények, tudnivalók

- A feladat elkészítésére 30 perc áll a rendelkezésre. Ez szigorú határidő, a Bíró előre megadott időben zár.
- A feladatokat számítógép előtt kell megoldani, tetszőleges fejlesztői környezetben, tetszőleges operációs rendszer segítségével.
- Az elkészült programot **20** alkalommal lehet benyújtani, a megadott határidőig.
- Csak a leírásban szereplő osztályokat, metódusokat és adattagokat kell megvalósítani, egyéb dolgokért nem jár plusz pont.
 - Aki Windowst használ, annak a gép elindítása után érdemes egyből a fejlesztőkörnyezetet elindítani, és létrehozni egy új projektet, és csak utána a böngészőt, mivel az elején egy néhány percig indexel, addig pont el lehet olvasni a feladatot.
- Bármely segédanyag használata **tilos** (a fejlesztőkörnyezetek nyújtotta segítségen kívül), aki mégis ilyen tesz, vagy próbálkozik vele, annak a dolgozata nem értékelhető és a ZH nem teljesített. Ha valakinek a padtársa segít, akkor mérlegelés nélkül mindkettő hallgató dolgozata sikertelen, a ZH nem teljesített.
- A feladat megoldása során minden megadott előírást pontosan követni kell! Tehát, ha a feladat leírása szerint egy adattag neve a "elsoFoku", akkor az alábbi elnevezések nem megfelelőek: "elsőFokú", "elsofoku", "elso_foku", "elsőFoq". Ugyanez igaz a metódusok, osztályok elnevezésére is!
- A metódusok esetében a visszatérési típus, a név, módosítók és a paraméterek típusai (és azok sorrendje) kerülnek ellenőrzésre, azonban a paraméterek nevei tetszőlegesek lehetnek.
- A Java elnevezési konvenciókat követni kell (getter/setter elnevezés, toString, indentálás, stb). Abban az esetben is, ha ezt a feladat külön nem emeli ki, az ellenőrzés során erre is építünk.
- A nem forduló kódok nem kerülnek kiértékelésre, ezt utólagosan a gyakorlatvezető sem bírálhatja felül. (Hiszen mindenki rendelkezésére áll a saját környezete, ahol fordítani, futtatni tudja a forráskódot, így feltöltés előtt ezt mindenképpen érdemes megnézni!)
- Az adattagok és konstruktorok hiányában garantáltan 0 pontos lesz a kiértékelés, mert ezek minden teszt alapját képezik.
- Ha végtelen ciklus van a programban, akkor ezt a Bíró ki fogja dobni 3 másodperc után (ha többször is meghívásra kerül ilyen metódus, akkor ez többszöri 3 másodperc, összesen akár 2 perc is lehet). Ilyenkor NE kattints még egyszer a *Feltöltés* gombra, mert akkor

kifagyhat a Bíró, csak a böngésző újraindításával lehet megoldani a problémát (emellett elveszik 1 feltöltési lehetőség is).

- Kérdés/probléma esetén a gyakorlatvezetők tudnak segítséget nyújtani.
- **A feladat megoldása során a default csomagba dolgozz, majd a kész forrásfájlokat tömörítve, zip formátumban töltsd fel, azonban a zip fájlt tetszőlegesen elnevezheted!**
 - Zip készítése: Windowson és Linuxon is lehet a GUI-ban jobb klikkes módszerrel tömörített állományt létrehozni (Windowsban pl. a 7-Zip nevű ingyenes program használatával).
 - Linux terminálon belül például a "zip feladat.zip *.java" paranccsal is elkészíthető a megfelelő állomány.
- A feladatokban az alábbi dolgok az alapértelmezettek (**kivéve**, ha a feladat szövege mást mond)
 - az osztályok láthatósága publikus
 - az egész érték 32 bites
 - a lebegőpontos számok dupla pontosságúak
 - az olyan metódusok void visszatéréssel rendelkeznek, amelyeknél nincs specifikálva visszatérési típus.
 - a metódusok mindenki számára láthatóak
 - az adattagok csak az adott osztályban legyenek elérhetőek
- A *riport.txt* és a fordítási log fájlok megtekinthetőek 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/IB204L-1/1/hXXXXXX/4/riport.txt`
 3. Az url-ből visszatörölve a 4-esig (*riport.txt* törlése) megkaphatók a 4-es próbálkozás adatai.
- Szövegek összehasonlításánál az egyezés a pontos egyezést jelenti, azaz ha kis-nagy betűben térnek el, akkor már nem tekinthetők egyenlőnek (pl. a "piros" != "Piros")
- A leírásokban bemutatott példákban a stringek 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 a 3 alma, de a szóköz szükséges!
- Az elvárt kimeneteknek karakterről karakterre olyan formátumúnak kell lennie, ami a feladatban le van írva (szóközöket és sortöréseket is beleértve).

1. ábra. Osztálydiagram



A feladat megoldása során használd a `minta.zip` fájlban lévő kiindulási projektet, amelyben megtalálható egy absztrakt osztály.

Suto osztály

Az osztály egy tetszőleges sütőt reprezentál, ahogy az osztálydiagramon is látható. Az osztályban található elkészített elemek:

- **tipus**: a sütő típusa,
- **etel**: a sütőben éppen lévő étel, kezdetben *null*,
- **koszossag**: a sütő koszossága, kezdetben *0*,
- **haviFogyasztas**: a sütő havi fogyasztása (egység), kezdetben *0*.

Az osztályban lévő metódusok:

- **eteltBerak**: étel berakása a sütőbe,
- **eteltKivesz**: a berakott étel kivétele a sütőből,
- **kitakarit**: a sütő kitakarítása,

- **sut**: a sütést reprezentáló absztrakt metódus,
- **koszossagNoveles**: a koszosság növelésére szolgáló absztrakt metódus.

Ezekhez a létező adattagokhoz, metódusokhoz tetszőleges új adattag/metódus hozzáadható, azonban a létező adattagok/metódusok módosítása (láthatóság, típus, egyéb módosítók, működés) nem engedélyezett (hibát dobhat a tesztelő).

A feladat két leszármazott osztály elkészítése: **Gazsuto**, **MikrohullamuSuto**.

Gazsuto osztály (5 pont)

A gázsütők speciális sütők, a szükséges új adattag az osztálydiagramon szerepel.

A **koszossagNoveles** metódus működése:

- Mivel a sütőben lévő koszok sütéskor egyre jobban ráégnék a sütő felületére, melynek takarítása nehezebb, ezért a metódus növelje meg a koszosságot az *aktuális koszosság * 2 + 1* értékre.

A **sut** metódus működése:

- Ha a paraméterből érkező fok nagyobb, mint a sütő maximális hőfoka, térjünk vissza hamissal.
- Egyéb esetben először betesszük az ételt az erre szolgáló *eteltBerak* metódussal.
- Növeljük a fogyasztást az alábbiak szerint: először, a paraméterben érkező hőfokra elő kell melegíteni a sütőt. A sütő 75 fokra olcsón felmelegszik, ennek költsége 10 fabatka. 75 °C felett a melegítés költsége 2 fabatka. Ezt követően pedig a szinten tartás 1 fabatka/perc. *Példa: 200 °C fokon szeretnénk sütni, 30 percig. Ekkor 75 fokra felmelegszik 10 fabatka költséggel, de mi 200 °C fokon szeretnénk sütni, tehát a maradék 125 °C hőfokra fokonként 2 fabatkát kell számolnunk (250 fabatka). Majd 30 percig sütünk, ami újabb 30 fabatka. Tehát a teljes költség 10 + 250 + 30 fabatka.* Minden esetben legalább 75 fokon akarunk sütni, ennél kisebb értékre nem kell felkészíteni a metódust.
- Növeljük meg a koszosságot az elkészített függvény segítségével.
- Végül kivesszük az ételt az erre szolgáló *eteltKivesz* metódussal.
- Sikeres sütés esetén térjünk vissza igazzal.

MikrohullamuSuto osztály (5 pont)

A mikrohullámú sütők speciális sütők, a szükséges új adattag az osztálydiagramon szerepel.

A **koszossagNoveles** metódus működése:

- Minden egyes mikrózáskor 1-gyel nő a koszosság.

A **sut** metódus működése:

- Növelje a havi fogyasztást az alábbiak szerint: a hőfokot először konvertáld át teljesítményre (double érték): $teljesitmeny = 3,27 * hőfok - 145,45$. Ha a teljesítmény nagyobb, mint az adott mikrohullámú sütő maximális teljesítménye, térj vissza hamissal.
- Egyéb esetben először betesszük az ételt az erre szolgáló *eteltBerak* metódussal.
- A teljesítményt (amit kiszámoltál) szorozd meg a sütési idővel. A kapott értéket oszd le 60-nal, az így kapott értékkel növeld a havi fogyasztást.
- Növeljük meg a koszoságot az elkészített függvény segítségével.
- Végül kivesszük az ételt az erre szolgáló *eteltKivesz* metódussal.
- Sikeres sütés esetén térjünk vissza igazzal.

Jó munkát!