

Java Programozás

Téma: **Zárthelyi dolgozat**, Java alapok

Tagozat / évfolyam: **nappali**, II. évfolyam

Dátum: **2013.04.11.** (április, csütörtök)

Kidolgozási idő: **90 perc**

Oktató: **Bedők Dávid** (qwaevisz@gmail.com)

Hallgató: _ _ _ _ _ (neptun kód) (név)

Készítsen egy olyan programot Eclipse IDE segítségével Java nyelven, mely az alábbi feladatra nyújt hatékony megoldást. A megvalósítás során a tanult eszközöket, módszereket és az elegáns, hatékony kód írott, íratlan szabályait használja. A programkód nyelve lehetőség szerint angol legyen, azonban magyar elnevezések használata során az ékezeteket és white spaceket mindenképpen kerülje el! A projektben használt java csomagok tartalmazzák a neptun kód és a teljes névnek ékezet és white space mentes kombinációját (pl. hu.q59r7a.bedokdavid). Nem forduló/futó alkalmazás nem kerül értékelésre.

Feltöltés: Interneten keresztül, kizárólag az óra végén a www.qwaevisz.hu weboldalon: **JAVA_20130411**.

Java alapismeretek (5 pont)

Készítsen konzol alkalmazást Java nyelven, mely egy **üzem munkafolyamatait** tartja karban, illetve monitorozza. Minden munkafolyamatra (abstract class **WorkProcess**, [**3 pont**]) jellemző az adott üzemben, hogy pontosan egy személy hajtja végre (class **Worker**), egy meghatározott üzemi területen (enum **Location**). Van egy állapota is (enum **State**), mely a következők egyike lehet: *Létrehozott* (Created), *Elkezdett* (Started) és *Befejezett* (Finished). Minden dolgozónak (class **Worker**, [**2 pont**]) legyen egy teljes neve (pl. *Nemecsek Ernő*), egy belső azonosítója (pl. *NEMERN*) és egy beosztása (enum **Role**). A Role (*Rakodó*, *Árufeltöltő*, *Osztályvezető*, stb.) és a Location (pl. *Raktár*, *Udvar*, stb.) enum-okat töltsük fel értelmes 4-5 értékkel.

Kétféle valós munkafolyamat létezik a feladat által definiált szimulációban: a gépesített (class **MechanizedProcess**, [**1 pont**]) és a pusztán kézzel végrehajtható (class **ManualProcess**, [**1 pont**]). A gépesített munkafolyamatban részt vesz egy gép is (enum **Machine**), míg a pusztán kézzel végrehajtható folyamatban tárolnunk kell a végrehajtási időt percben (int duration).

Megjegyzés: mielőtt továbblép a következő részfeladatokra, bizonyosodjon meg róla, hogy az elkészült osztályai helyesen működnek, az asszociációk rendben vannak, stb. Részletes toString() metódusokat mindenhol készítsen! A tesztek [**3 pont**]-ot pontot, míg a toString()-ek összesen [**2 pont**]-ot érnek.

Ha eddig minden jó, akkor már közel a gyenge eredmény határa!

A munkafolyamatok vezérlésére készítse el a következő metódusokat:

- Minden munkafolyamat miután létrejött, automatikusan *Létrehozott* állapotba kerül (ctor).
- Ha egy **String start()** metódusát meghívjuk, akkor a következők egyikét adja kimenetül (a kimenet mellett a munkafolyamat állapotát (ha indokolt) akkor *Létrehozott*-ból *Elkezdett*-re állítja):

- [személy azon.] elindított egy munkafolyamatot [itt]. A folyamat [időtartam] ideig fog tartani.
- [személy azon.] elindított egy munkafolyamatot [itt]. A folyamatban résztvesz egy [munkagép] is.
- [személy azon.] megpróbált egy már befejezett munkafolyamatot elindítani.
- Ha egy **String end()** metódusát meghívjuk, akkor a következők egyikét adja kimenetül (a kimenet mellett a munkafolyamat állapotát (ha indokolt) akkor *Elkezdett*-ről *Befejezett*-re állítja):
 - [személy azon.] befejezett egy munkafolyamatot [itt]. A folyamat [időtartam] ideig tartott.
 - [személy azon.] befejezett egy munkafolyamatot [itt]. A folyamatban résztvevő [munkagép] felszabadult.
 - [személy azon.] megpróbált egy már befejezett munkafolyamatot lezárni.
 - [személy azon.] megpróbált egy még elindítatlan munkafolyamatot lezárni.

A `start()` és `end()` metódusok implementálása során használjon `protected abstract` metódusokat, de a `start()` és `end()` metódusok az `ősben` legyenek (redundás részek kiemelése végett)! Más megoldás nem elfogadott ([**5 pont**]).

Megjegyzés: ezen a ponton ne menjen addig tovább, amíg nem készít teszteként mindkét munkafolyamat minden sikeres esetére nézve, illetve legyen tesztelve a három sikertelen eset is (vagyis legalább a fenti 7 kimenet mindegyike jelenjen meg a konzol képernyőjén) [**3 pont**].

Ha eddig minden közel jó, akkor már biztos közepes eredménye van!

Készítsen egy `Vállalat` osztályt (class **Company**), mely munkafolyamatok generikus listáját tartalmazza és kezelje. Legyen egy **void addWorkProcess(...)** metódusa, mely eltárol egy tetszőleges munkafolyamatot. Az osztály `ToString()` metódusa listázza ki az összes munkafolyamatot.

Megjegyzés: addig ne menjen tovább, amíg a `Vállalat` osztály feltöltését és megjelenítését nem tesztelte [**2 pont**]!

Ha eddig minden jó, akkor már biztos jó eredménye van!

Készítsen a `Vállalat` osztályba egy **String start(Location location)** metódust, mely a megadott üzemi terület összes elindítható munkafolyamatát elindítja! Térjen vissza az elindított munkafolyamatok kiemelteinek összefűzött értékével (*opcionális*: `List<String>`-el térjen vissza) [**2 pont**]!

Készítsen a `Vállalat` osztályba egy **String end(Role role)** metódust, mely a megadott szerepkörrel rendelkező összes dolgozó összes futó folyamatát lezárja. Térjen vissza a lezárt folyamatok kimeneteinek összefűzött értékével [**3 pont**].

Összesen: 25 pont, gyenge 13 ponttól.