

Requirement

System definition

A program célja, hogy képes legyen feladatokat betölteni, és azokat időkorláton belül elvégeztetni a felhasználóval. Ellenőriznie kell a megoldásokat, és annak megfelelősége alapján fel kell hívnia a felhasználót egy kötelességre (ivás). Amennyiben a megoldás helyes, engedélyezhet neki további feladatokat, melyek előkövetelménye volt, ez a feladat.

Programnak amellet, hogy jegyeznie kell az eddig megvalósított feladatokat, tárolnia kell, hogy a felhasználó eddig milyen kötelességeket kellett, hogy teljesítsen. Ezekből az adatokból további érdekességeket számolhat ki (pl.: megivott alkohol mennyisége, testsúly, testmagasság -> véralkoholszint).

A programot több gépen futtatva párhuzamosan esélykiegyenlítő programozóversenyt lehet tartani. Az abszolút nyertes az, akinek a feladatok megoldásához szükséges fennmaradó idejének az összege a legtöbb. Illetve citromdíjas a legmagasabb alkoholszintű lehet.

A hozott alkohol milyensége, mennyisége a program (verseny) indításakor felhasználóra szabva, meg kell adni, és a program majd abból ajánl fel.

A programhoz tetszőleges számú fordító hozzáadható java nyelven. Alapból a C, C++ és Java fordító lesz, de ez bővíthető továbbiakkal.

Project plan

Specification: Megírjuk a specifikációt, benne az elvárásokkal és a User interface leírásával. Ezzel együtt megírunk egy kezdetleges user's manualt, és a verifikációs tervet – melyik fázist mikor tekintjük befejezettnek.

Architectural design: Majd elkezdjük tervezni kezdetlegesen a programot, egy magasabb absztrakciós szintről. Készítünk egy magasabb absztrakciós szintű UML class diagramot, és egy állapot diagramot. Leírjuk az egyes osztályok kapcsolatát és feladatát.

Detailed design: UML diagrammon kifejtjük az egyes osztályokat adattagjaikkal és metódusaikkal. Kapcsolataikat és feladatukat részletesebben leírjuk. Továbbá leírjuk a program futását, röviden lerajzoljuk egy szekvencia diagrammal. Megírjuk a részletes user's manualt, és a verifikációs tervet.

Implementation: megírjuk a kódot Java nyelven. Az egyes kritikusabb elemeket teszteljük. Esetleges változtatás esetén frissítjük a dokumentációkat. Amint a program megfelelően működik, további optimalizálást eszközölünk. Kellőképp kommentelt kódból generálunk egy javadoc-ot.

Validation: utolsó alkalommal, változtatások követésével frissítjük a dokumentációkat és levonjuk a következtetéseket, tanulságokat a projekt menetét illetően. A különböző méréseket, amiket a projekt során végrehajtottunk, ellenőrizzük és interpretáljuk.