

# Specification

## Requirement specification

Elvárások a programtól:

- feladatok (serializált java objektumok) betöltése, feldolgozása, játszátása
- C, C++, Java (és opcionálisan későbbi verziókban további nyelvek) beolvasása (program felületén írva), lefordítása, standard inputra, outputra írás/olvasás, eredmény ellenőrzése
- feladatmegoldás idejének, programfutás idejének, alkoholszint mérése
- különböző fajtájú, alkoholszázalékú italok tárolása felhasználóra szabva mennyiséggel és azokból különböző eseményeket, követően különböző mennyiségek (átváltás kortyra) elfogyasztására felhívás
- elvégzett feladatok tárolása, feladatmegoldás- és futásidővel; a nem elvégzett feladatokra épülő további feladatok elkezdésének gátlása

A GUI felületen szerepeljen:

- új játék kezdése gomb (confirmation ablakkal)
- programíró részleg
  - nyelv választó legördülő menü aktuális feladatra
  - szövegszerkesztő, program írására
  - beküldő gomb
- felhasználó részleg:
  - név
  - nem
- alkohol részleg:
  - új ital felvétele gomb (frissíthetünk előző megadott italok mennyiségét is)
  - ivásra rendelkező italok alkoholszázaléka, mennyisége (megivott, hátramaradt)
  - italonként egy gomb, hogy abból iszik
  - véralkoholszint
- megoldások részleg:
  - *megoldás idők*
  - *futásidők*
  - *hibás beküldések száma*
  - *nyelv*
  - hátramaradt pluszidők összegezve
  - átlagos futásidő
  - helyes beküldések aránya
  - megoldták-e
- aktuális feladat részleg:
  - új feladat betöltése gomb
  - feladat szövege

- feladat amire épül
- felhasználható megoldási idő
- eltelt idő a feladat betöltésétől
- alkohommennyiség fogyasztásra különböző kimeneteleknél
- eddigi hibás beküldések száma és a megengedett
- előző megoldás kimenetele

## Preliminary user's manual

Új játék kezdéséhez, először rákattintunk az *Új játék* gombra, majd megadjuk a felugró ablakban az adatainkat, mint: név, nem, testsúly, testtömeg. Nyomunk egy OK-t, így visszatérünk a főablakra.

A játék nem kezdhető el, amíg nem rendelkezünk alkohollal. Így következő lépésként, adjunk hozzá italokat az alkoholrészlegnél: nyomjuk meg az *Új ital felvétele* gombot, majd a felugró ablakban adjuk meg milyen italokkal rendelkezünk – név, mennyiség, alkoholszázalék.

Amint ezzel megvoltunk, vegyük fel az első feladatot az aktuális feladat részlegnél: nyomjuk meg az *Új feladat betöltése* gombot, könyvtárrendszerben keressük meg a kívánt feladatot és nyissuk meg. Ezután feljön egy újabb ablak, amin szerepel ha megfelelő mennyiségű alkoholunk van e a feladat elkezdéséhez, illetve szerepel rajta, hogy mennyit hibázhatunk még, hogy elegendő italunk maradjon a befejezéshez. Ezen az ablakon, amint megnyomjuk az *OK* gombot (amennyiben megfelelő mennyiségű itallal rendelkezünk), elindul az időzítő, megjelenik a feladat szövege az aktuális feladat részlegben és aktiválódik a nyelv kiválasztó legördülő menü.

Ezután válasszuk ki a kívánt nyelvet a legördülő menüből, így aktiválódik a szövegszerkesztő és a *Beküldés* gomb. A szövegszerkesztőbe írjuk a megoldásunkat, majd ha jónak találjuk, kattintsunk rá a *Beküldés* gombra. Amint rákattintottunk, az aktuális feladat részlegben megjelenik az eredmény és hogy mennyit kell innunk. Amennyiben ki van írva alkohommennyiség, választanunk kell a rendelkezésünkre álló alkoholokból, hogy azt melyikből fogyasztjuk el, rányomva az ital melletti *Ebből iszok* gombra. Amíg ezt nem tesszük meg, nem haladhatunk tovább, akár az aktuális, akár a következő feladat megoldásában.

A további feladatok megoldását, addig folytathatjuk, amíg van elég italunk/feladatunk.

## Premilinary Verification plan

**Architectural design:** végeztünk vele ha megvan egy UML class diagram (uml\_class\_abstract), és egy állapot diagram (uml\_state\_abstract). Ezen felül leírjuk a főbb osztályok feladatát és kapcsolatát (main\_classes\_abstract.doc). Requirement és specification frissítése.

**Detailed design:** kifejtett UML class diagram (uml\_class\_detailed), osztályok részletes leírása és kapcsolata (classes\_detailed.doc), program futásának leírása és ábrázolása szekvencia diagramon (run.doc, run\_sequence), részletes user's manual (user\_manual\_detailed.doc), részletes verifikációs terv (verification\_plan\_detailed.doc). Requirement, specification és az architectural design frissítése.

**Implementation:** Java program optimalizálással, Java tesztek, részletes kommentelés a kódban, Javadoc generálása. Requirement, specification, designok frissítése.

**Validation:** minden doksit átnézni és frissíteni. Levonni a következtetéseket (validation.doc).