# Tesztelési dokumentáció

# Programozás I TG

# IR111

# Dinnyés Balázs; Keleti Márton; Somogyvári Kristóf

# 1. GUI tesztelése

|  |
| --- |
| Első tesztkör: |
| A elérendő cél: a bejelentkező, információs ablak és a főprogram panel elemeinek helyes működése, megjelenítése. |
| A módosítás megtörtént, következik a tesztelés |
| teszt eredmény: A három kép, együtt sok az információs ablakon, és nem jelentek meg helyesen, az ablak zsúfolt; a forgó, díszítő elem nem jelenik meg teljes egészében; a színek a rádiógombok feliratán nem eléggé különbözőek |
| a javítás célja: maradjon 1 kép a bejelentkező információs ablakon: ez a html logó -> legyen középen; a forgó földgömb maradhat a helyén, így is elég látványos; színek megadása; |
| javítás megtörtént, cél elérve, eredmények helyesek |

|  |
| --- |
| második tesztkör: |
| A elérendő cél: A főprogramban egy JLabel címke értesítsen a konvertált és az eredmény file nevéről, így a művelet után is van információ róluk; az átkonvertált köztes dokumentumok legyenek lementve file - ban |
| A módosítás megtörtént, következik a tesztelés |
| input: a címkén a háromféle file neve (wiki, md, html) |
| output: a címkén: eredmény.txt |
| teszt eredmény: a nevek a címkén nem eléggé informatívak, nincs elég space karakter a szavak közt; a címke túl hosszú, kilóg a látótérből; |
| a javítás célja: a program indulásakor a címke legyen inicializálva szöveggel; a nevek utaljanak a művelet eredményére pl: in.Html - > in\_out.Html és minimalizálni a szavak számát, a file kiválasztó ablaknak legyen all – file funkciója is, így mindhárom féle file - t lehessen közvetlenül kiválasztani |
| javítás megtörtént, cél elérve, eredmények helyesek, out: a példa szerintiek; |

# 2. Markdown átalakítás tesztelése

|  |
| --- |
| Első tesztkör: |
| A elérendő cél: a markdown bemenetet html-re alakítsuk adatvesztés nélkül |
| Implementálás külső, nyílt forrású könyvtárral megtörtént, következik a tesztelés |
| teszt eredmény: a kimenetben minden adat megtalálható volt, azonban bizonyos formázások nem a kívánt eredményt produkálták |
| a javítás célja: lehető legtöbb formázás helyes átalakítása |
| A markdown parsert lecserértük, az eredmény javult. |

# 3. Wiki átalakítás tesztelése

|  |
| --- |
| Első tesztkör: |
| A elérendő cél: a wikitext bemenetet html-re alakítsuk adatvesztés nélkül |
| Implementálás ugyanazzal a külső könyvtárral megtörtént, következik a tesztelés |
| teszt eredmény: a kimenetben minden adat megtalálható volt, azonban a <ref> tagek is megjelennek a végső kimenetben |
| a javítás célja: <ref> tagek eltüntetése |
| A html parserben lekezeljük, hogy az ismeretlen html tageket tüntesse el, a javítás sikeres. |

# 4. HTML átalakítás tesztelése

|  |
| --- |
| Első tesztkör: |
| A elérendő cél: egy nem agyonbonyolított html fájl rendezett txt kimenetre konvertálása |
| Implementálás jsoup külső könyvtár használatával, fabejárással, saját parser metódusban megtörtént. |
| teszt eredmény: a kimenetben minden adat megtalálható volt, formai csiszolások, táblázat implementálása volt szükséges |
| a javítás célja: táblázatok is jelenjenek meg a kimenetben |
| A modelben 3 új osztály a táblázat, annak sorai és celláinak kezelésére. Megkülönböztetünk normál <td> és fejléc <th> cellákat. Amennyiben egy sorban legalább egy fejléc cella található, az adott sort teljes egészében aláhúzzuk, ezáltal elkülönítve a többitől. A javítás elérte a kívánt eredményt. |