



Szoftver projekt laboratórium

X. ÖSSZESÍTETT DOKUMENTUM

Csapat

Csapatorszám - Csapatnév

Konzulens

Dr. Goldschmidt Balázs

Csapattagok

Teszt Tamás	NEPTUN	teszttamas@iit.bme.hu
Java Dániel	??????	javadaniel@iit.bme.hu
Projlab Péter	??????	projlabpeter@iit.bme.hu
LaTeX Zoltán	??????	latexzoltan@iit.bme.hu
Git Attila	??????	gitattila@iit.bme.hu

2021. február 2.

2. fejezet

Követelmény, projekt, funkcionalitás

2.1. Bevezetés

2.1.1. Cél

A dokumentum célja.

2.1.2. Szakterület

A kialakítandó szoftver milyen területen használható, milyen célra.

2.1.3. Definíciók, rövidítések

A dokumentumban használt definíciók, rövidítések magyarázata.

1. Definíció. *Példa defi: egy demó a defi használatára*

1. Rövidítés. *PR: példa rövidítés*

2.1.4. Hivatkozások

► BME IIT - Programozás alapjai 3. segédanyagok, Szoftvertechnikák segédanyagok, Szoftver projekt laboratórium feladatok - <http://iit.bme.hu/>

2.1.5. Összefoglalás

A dokumentum további részeinek rövid ismertetése

2.2. Áttekintés

2.2.1. Általános áttekintés

A kialakítandó szoftver legmagasabb szintű architektúráis képe. A fontosabb alrendszerek felsorolása, a közöttük kialakítandó interfészek lényege, a felhasználói kapcsolatok alapja. Esetleges hálózati és adattárolási elvárások.

2.2.2. Funkciók

A feladat kb. 4000 karakteres (kb 1,5 oldal) részletezettségű magyar nyelvű leírása.
Nem szerepelhetnek informatikai kifejezések.

2.2.3. Felhasználók

A felhasználók jellemzői, tulajdonságai

2.2.4. Korlátozások

Az elkészítendő szoftverre vonatkozó – általában nem funkcionális - előírások, korlátozások.

2.2.5. Feltételezések, kapcsolatok

A Hivatkozásokban felsorolt anyagok, web-oldalak kapcsolódása a feladathoz, melyik milyen szempontból érdekes, milyen inputot ad.

2.3. Követelmények

2.3.1. Funkcionális követelmények

Az alábbi táblázat kitöltésével készítenő. Dolgozzon ki követelmény azonosító rendszert! Az ellenőrzés módja szokásosan bemutatás és/vagy kiértékelés. Prioritás lehet alapvető, fontos, opcionális. Az alapvető követelmények nem teljesítése végzetes. Forrás alatt a követelményt előíró anyagot, szervezetet kell érteni. Esetünkben forrás lehet maga a csapat is, mikor ő talál ki követelményt. Use-case-ek alatt az adott követelményt megvalósító használati esete(ke)t kell megadni.

Azonosító	Prioritás	Forrás	Use-case	Ellenőrzés
Azonosító	Pioritás	Forrás	Használati eset	llenőrzés

Leírás: Leírás

Megjegyzés (opcionális)

2.3.2. Erőforrásokkal kapcsolatos követelmények

A szoftver fejlesztésével és használatával kapcsolatos számítógépes, hardveres, alapszoftveres és egyéb architekturális és logisztikai követelmények

Azonosító	Prioritás	Forrás	Ellenőrzés
Azonosító	Pioritás	Forrás	Ellenőrzés

Leírás: Leírás

Megjegyzés (opcionális)

2.3.3. Átadással kapcsolatos követelmények

A szoftver átadásával, telepítésével, üzembe helyezésével kapcsolatos követelmények

Azonosító	Prioritás	Forrás	Ellenőrzés
Azonosító	Pioritás	Forrás	Ellenőrzés

Leírás: Leírás

Megjegyzés (opcionális)

2.3.4. Egyéb nem funkcionális követelmények

A biztonsággal, hordozhatósággal, megbízhatósággal, tesztelhetőséggel, a felhasználóval kapcsolatos követelmények

Azonosító	Prioritás	Forrás	Ellenőrzés
Azonosító	Pioritás	Forrás	Ellenőrzés

Leírás: Leírás

Megjegyzés (opcionális)

2.4. Lényeges use-case-ek

Funkcionális követelmények részben felsorolt követelmények közül az alapvető és fontos követelményekhez tartozó használati esetek megadása az alábbi táblázatos formában.

2.4.1. Use-case leírások

Minden use-case-hez külön

Use-case neve:	USE-CASE NEVE
Rövid leírás:	Az eset rövid leírása
Aktorok:	Aktorok
Forgatókönyv:	Forgatókönyv A.1 Alternatíva

2.4.2. Use-case diagram



2.1. ábra. Demó

2.5. Szótár

A szótár a követelmények alapján készítendő fejezet. Egy szótári bejegyzés definiálásához csak más szótári bejegyzések és köznapi – a feladattól független – fogalmak használhatók fel. A szótár mérete kb. 1-2 oldal legyen. A bejegyzések legyenek ABC sorrendben!

Kulcs 1	Érték 1
Hóásás	Egy réteg hó eltakarítása, azaz törlése az adott jégtábláról. Egy egység munkába kerül.

2.6. Projekt terv

Tartalmaznia kell a projekt végrehajtásának lépéseit, a lépések, eredmények határidejét, az egyes feladatok elvégzéséért felelős személyek nevét és beosztását, a szükséges erőforrásokat, stb. Meg kell adni a csoportmunkát támogató eszközöket, a választott technikákat! Definiálni kell, hogy hogyan történik a dokumentumok és a forráskód megosztása!

2.6.1. Projektütemterv

Határidő	Feladat	Felelős
febr. 4.	Követelmény, projekt, funkcionalitás	??????
márc. 2.	Analízis modell kidolgozása 1. - beadás	??????
márc. 9.	Analízis modell kidolgozása 2. - beadás	??????
márc. 16.	Skeleton tervezése - beadás	??????
márc. 23.	Skeleton - beadás és a forráskód herculesre való feltöltése	??????
márc. 30.	Prototípus koncepciója - beadás	??????
ápr. 6.	Részletes tervek - beadás	??????
ápr. 27.	Prototípus - beadás és a forráskód, a teszt-bemenetek és az elvárt kimenetek herculesre való feltöltése	??????
máj. 4.	Grafikus felület specifikációja - beadás	??????
máj. 18.	Grafikus változat és Összefoglalás - beadás és a forráskód herculesre való feltöltése	??????

2.6.2. Erőforrások, eszközök

A fejlesztés során felhasznált segédeszközök:

- Dokumentáció: dokumentáló eszközök felsorolása
- Kommunikáció: kommunikációs platformok...

- Modellező eszköz: modellezési eszközök
- Fejlesztő környezetek: ...
- Forráskód megosztás, verziókezelés: ...
- Egyéb

Még szabadon felvehető releváns idetartozó dolgok...

2.7. Napló

A napló tartalmazza az előző beadás óta eltelt időszak történéseit időrendben. A naplóból egyértelműen ki kell derülnie, hogy az egyes anyagrészeket ki és mennyi idő alatt készítette.

A napló bejegyzésekből áll. Minden bejegyzésnek tartalmaznia kell:

- a történes kezdetének időpontját, nap-óra pontossággal
- történes időtartamát, óra felbontással
- a szereplő(k) nevét (Kérjük a szereplők VEZETÉKNEVÉT használni)
- a tevékenység leírását.

Amennyiben a tevékenységben több szereplő vesz részt, akkor aza tevékenység csak értekezlet lehet, amelynek az eredményei DÖNTÉSEK. A döntéseket precízen meg kell szövegezni (Pl.: Az X objektum Y és Z metódusainak kódját W készíti el Q határidőre). Ha a bejegyzés egyetlen személyhez kötődik, akkor meg kell adni, hogy a tevékenység milyen dologra irányul. A dolog a feladat kapcsán elkészítendő termék, amelynek a (esetleg korábban) beadott anyagban megtalálhatónak kell lenni. A naplóbejegyzés felbontásának egysége szöveges, rajzos anyag esetében az ábra, diagram, vagy kb. fél-egy oldalnyi szöveg. Kódban az egység a metódus. (Pl.: A 3. ábrán látható szekvencia-diagram kidolgozása, vagy az X objektum Y és Z metódusainak kódolása és belövése.

Kezdet	Időtartam	Résztvevők	Leírás
feb. 18. 16h	1 óra	Csapat	Értekezlet Döntés: Segédeszközök kiválasztása (git, trello, drive)

3. fejezet

Analízis modell kidolgozása 1.

3.1. Objektum katalógus

Minden, a feladatban szereplő objektum rövid, egy-két bekezdés hosszú ismertetése. Meg kell jelenjen minden objektumhoz, hogy mi a felelőssége. Informális leírás, ezért nem kell foglalkozni az örökléssel, az interfészekkel, az absztrakt osztályokkal, a segédosztályokkal.

3.1.1. Objektum1

Felelősség informális leírása

3.1.2. Objektum2

Felelősség informális leírása

Az objektumkatalógus alapján kiindulva kell megalkotni az objektumorientált analízis modellt. A 3.2, 3.3, 3.4, 3.5 alfejezetek ugyanannak a modellnek a különböző nézetei, ezért egy időben, egymással összefüggésben készülnek. Megtörténik az objektumkatalógusbantárgyalt objektumok felelősségének formalizálása osztályokká, attribútumokká, metódusokká. Csak publikus metódusok szerepelhetnek. Megjelennek az interfészek, az öröklés, az absztrakt osztályok. Segédosztályokra még nincs szükség.

3.2. Statikus struktúra diagramok

Az előző alfejezet osztályainak kapcsolatait és publikus metódusait bemutató osztálydiagram(ok). Tipikus hibalehetőségek: csillag-topológia, szigetek.



3.1. ábra. Demó

3.3. Osztályok leírása

Az előző alfejezetben tárgyalt objektumok felelősségének formalizálása attribútumokká, metódusokká. Csak publikus metódusok szerepelhetnek. Ebben az alfejezetben megjelennek az interfészek, az öröklés, az absztrakt osztályok. Segédosztályokra még mindig nincs szükség. Az osztályok ABC sorrendben kövessék egymást. Interfészek esetén az Interfészek, Attribútumok pontok kimaradnak.

3.3.1. Osztály1

■ Felelősség

Felelősség leírása

■ Interfészek

Megvalósított interfészek felsorolása

■ Ősosztály

Ős-Ősosztály \rightarrow Ősosztály...

■ Attribútumok

$\diamond +A [0..*]$ - adattag A

Milyen publikus, protected és privát metódusokkal rendelkezik. Metódusonként egy-három mondat arról, hogy mit csinál.

■ Metódusok

$\diamond +B(A \ a) : void$ - metódus B

3.3.2. Osztály2

■ Felelősség

Felelősség leírása

■ Állapotgép

A belső működéshez tartozó állapotgép



3.2. ábra. Az osztály állapotváltozásai

3.4. Statikus struktúra diagramok

Az előző alfejezet osztályainak kapcsolatait és publikus metódusait bemutató osztálydiagram(ok). Tipikus hibalehetőségek: csillag-topológia, szigetek.



3.3. ábra. osztálydiagram

3.5. Szekvencia diagramok

Inicializálásra, use-case-ekre, belső működésre. Konzisztens kell legyen az előző alfejezettel. Minden metódus, ami ott szerepel, fel kell tűnjön valamelyik szekvenciában.

Minden metódusnak, ami szekvenciában szerepel, szereplnie kell a valamelyik osztálydiagramon.



3.4. ábra. Szekvencia1

3.6. State-chartok

Csak azokhoz az osztályokhoz, ahol van értelme. Egyetlen állapotból álló state-chartok ne szerepeljenek. A játék működését bemutató state-chart-ot készíteni tilos.

3.7. Napló

A napló tartalmazza az előző beadás óta eltelt időszak történéseit időrendben. A naplóból egyértelműen ki kell derülnie, hogy az egyes anyagrészeket ki és mennyi idő alatt készítette.

A napló bejegyzésekből áll. Minden bejegyzésnek tartalmaznia kell:

- a történes kezdetének időpontját, nap-óra pontossággal
- történes időtartamát, óra felbontással
- a szereplő(k) nevét (Kérjük a szereplők VEZETÉKNEVÉT használni)
- a tevékenység leírását.

Amennyiben a tevékenységben több szereplő vesz részt, akkor aza tevékenység csak értekezlet lehet, amelynek az eredményei DÖNTÉSEK. A döntéseket precízen meg kell szövegezni (Pl.: Az X objektum Y és Z metódusainak kódját W készíti el Q határidőre). Ha a bejegyzés egyetlen személyhez kötődik, akkor meg kell adni, hogy a tevékenység milyen dologra irányul. A dolog a feladat kapcsán elkészítendő termék, amelynek a (esetleg korábban) beadott anyagban megtalálhatónak kell lenni. A naplóbejegyzés felbontásának egysége szöveges, rajzos anyag esetében az ábra, diagram, vagy kb. fél-egy oldalnyi szöveg. Kódban az egység a metódus. (Pl.: A 3. ábrán látható szekvencia-diagram kidolgozása, vagy az X objektum Y és Z metódusainak kódolása és belövése.

Kezdet	Időtartam	Résztvevők	Leírás
feb. 18. 16h	1 óra	Csapat	Értekezlet Döntés: Segédeszközök kiválasztása (git, trello, drive)

4. fejezet

Analízis modell kidolgozása 2.

4.1. Objektum katalógus

Minden, a feladatban szereplő objektum rövid, egy-két bekezdés hosszú ismertetése. Meg kell jelenjen minden objektumhoz, hogy mi a felelőssége. Informális leírás, ezért nem kell foglalkozni az örökléssel, az interfészekkel, az absztrakt osztályokkal, a segédosztályokkal.

4.1.1. Objektum1

Felelősség informális leírása

4.1.2. Objektum2

Felelősség informális leírása

Az objektumkatalógus alapján kiindulva kell megalkotni az objektumorientált analízis modellt. A 3.2, 3.3, 3.4, 3.5 alfejezetek ugyanannak a modellnek a különböző nézetei, ezért egyidőben, egymással összefüggésben készülnek. Megtörténik az objektumkatalógusbantárgyalt objektumok felelősségének formalizálása osztályokká, attribútumokká, metódusokká. Csak publikus metódusok szerepelhetnek. Megjelennek az interfészek, az öröklés, az absztrakt osztályok. Segédosztályokra még nincs szükség.

4.2. Statikus struktúra diagramok

Az előző alfejezet osztályainak kapcsolatait és publikus metódusait bemutató osztálydiagram(ok). Tipikus hibalehetőségek: csillag-topológia, szigetek.



4.1. ábra. Demó

4.3. Osztályok leírása

Az előző alfejezetben tárgyalt objektumok felelősségének formalizálása attribútumokká, metódusokká. Csak publikus metódusok szerepelhetnek. Ebben az alfejezetben megjelennek az interfészek, az öröklés, az absztrakt osztályok. Segédosztályokra még mindig nincs szükség. Az osztályok ABC sorrendben kövessék egymást. Interfészek esetén az Interfészek, Attribútumok pontok kimaradnak.

4.3.1. Osztály1

■ Felelősség

Felelősség leírása

■ Interfészek

Megvalósított interfészek felsorolása

■ Ősosztály

Ős-Ősosztály \rightarrow Ősosztály...

■ Attribútumok

$\diamond +A [0..*]$ - adattag A

Milyen publikus, protected és privát metódusokkal rendelkezik. Metódusonként egy-három mondat arról, hogy mit csinál.

■ Metódusok

$\diamond +B(A\ a) : void$ - metódus B

4.3.2. Osztály2

■ Felelősség

Felelősség leírása

■ Állapotgép

A belső működéshez tartozó állapotgép



4.2. ábra. Az osztály állapotváltozásai

4.4. Statikus struktúra diagramok

Az előző alfejezet osztályainak kapcsolatait és publikus metódusait bemutató osztálydiagram(ok). Tipikus hibalehetőségek: csillag-topológia, szigetek.



4.3. ábra. osztálydiagram

4.5. Szekvencia diagramok

Inicializálásra, use-case-ekre, belső működésre. Konzisztens kell legyen az előző alfejezettel. Minden metódus, ami ott szerepel, fel kell tűnjön valamelyik szekvenciában.

Minden metódusnak, ami szekvenciában szerepel, szereplnie kell a valamelyik osztálydiagramon.



4.4. ábra. Szekvencia1

4.6. State-chartok

Csak azokhoz az osztályokhoz, ahol van értelme. Egyetlen állapotból álló state-chartok ne szerepeljenek. A játék működését bemutató state-chart-ot készíteni tilos.

4.7. Napló

A napló tartalmazza az előző beadás óta eltelt időszak történéseit időrendben. A naplóból egyértelműen ki kell derülnie, hogy az egyes anyagrészeket ki és mennyi idő alatt készítette.

A napló bejegyzésekből áll. Minden bejegyzésnek tartalmaznia kell:

- a történes kezdetének időpontját, nap-óra pontossággal
- történes időtartamát, óra felbontással
- a szereplő(k) nevét (Kérjük a szereplők VEZETÉKNEVÉT használni)
- a tevékenység leírását.

Amennyiben a tevékenységben több szereplő vesz részt, akkor aza tevékenység csak értekezlet lehet, amelynek az eredményei DÖNTÉSEK. A döntéseket precízen meg kell szövegezni (Pl.: Az X objektum Y és Z metódusainak kódját W készíti el Q határidőre). Ha a bejegyzés egyetlen személyhez kötődik, akkor meg kell adni, hogy a tevékenység milyen dologra irányul. A dolog a feladat kapcsán elkészítendő termék, amelynek a (esetleg korábban) beadott anyagban megtalálhatónak kell lenni. A naplóbejegyzés felbontásának egysége szöveges, rajzos anyag esetében az ábra, diagram, vagy kb. fél-egy oldalnyi szöveg. Kódban az egység a metódus. (Pl.: A 3. ábrán látható szekvencia-diagram kidolgozása, vagy az X objektum Y és Z metódusainak kódolása és belövése.

Kezdet	Időtartam	Résztvevők	Leírás
feb. 18. 16h	1 óra	Csapat	Értekezlet Döntés: Segédeszközök kiválasztása (git, trello, drive)

5. fejezet

Szkeleton tervezése

5.1. A szkeleton modell valóságos use-case-ei

A szkeletonnak, mint önálló programnak a működésével kapcsolatos use-case-ek.

5.1.1. Use-case diagram



5.1. ábra. Demó

5.1.2. Use-case leírások

Minden use-case-hez külön

Use-case neve:	USE-CASE NEVE
Rövid leírás:	Az eset rövid leírása
Aktorok:	Aktorok
Forgatókönyv:	Forgatókönyv A.1 Alternatíva

5.2. A szkeleton kezelői felületének terve, dialógusok

A szkeleton által elfogadott bemenetek , valamint a szöveges konzolon megjelenő kimenetek. A kiemenet formátuma olyan kell legyen, ami alapján a működés összevethető a korábbi szekvencia-diagramokkal.

Konzol input/output szemléltetésre:

```
move 1 to 3  
get item
```

5.3. Szekvencia diagramok a belső működésre

A szkeletonban implementált szekvenciadiagramok. Tipikusan egy use-case egy diagram. Ezek megegyezhetnek a korábban specifikált diagramokkal, de az egyes életvonalakat (lifeline) egyértelműen a szkeletonban példányosított objektumokhoz kell tudni kötni. Azt kell megjeleníteni, hogy a szkeletonban létrehozott objektumok egymással hogyan fognak kommunikálni.

5.4. Kommunikációs diagramok

A szkeletonban, az egyes szkeleton-use-case-ek futása során létrehozott objektumok és kapcsolataik bemutatására szolgáló diagramok. Ezek alapján valósítják meg a szkeleton fejlesztői az inicializáló kódrészleteket.

5.5. Napló

A napló tartalmazza az előző beadás óta eltelt időszak történéseit időrendben. A naplóból egyértelműen ki kell derülnie, hogy az egyes anyagrészeket ki és mennyi idő alatt készítette.

A napló bejegyzésekből áll. Minden bejegyzésnek tartalmaznia kell:

- a történes kezdetének időpontját, nap-óra pontossággal
- történes időtartamát, óra felbontással
- a szereplő(k) nevét (Kérjük a szereplők VEZETÉKNEVÉT használni)
- a tevékenység leírását.

Amennyiben a tevékenységben több szereplő vesz részt, akkor aza tevékenység csak értekezlet lehet, amelynek az eredményei DÖNTÉSEK. A döntéseket precízen meg kell szövegezni (Pl.: Az X objektum Y és Z metódusainak kódját W készíti el Q határidőre). Ha a bejegyzés egyetlen személyhez kötődik, akkor meg kell adni, hogy a tevékenység milyen dologra irányul. A dolog a feladat kapcsán elkészítendő termék, amelynek a (esetleg korábban) beadott anyagban megtalálhatónak kell lenni. A naplóbejegyzés felbontásának egysége szöveges, rajzos anyag esetében az ábra, diagram, vagy kb. fél-egy oldalnyi szöveg. Kódban az egység a metódus. (Pl.: A 3. ábrán látható szekvencia-diagram kidolgozása, vagy az X objektum Y és Z metódusainak kódolása és belövése.

Kezdet	Időtartam	Résztvevők	Leírás
feb. 18. 16h	1 óra	Csapat	Értekezlet Döntés: Segédeszközök kiválasztása (git, trello, drive)

6. fejezet

Szkeleton beadás

6.1. Fordítási és futtatási útmutató

A feltöltött program fordításával és futtatásával kapcsolatos útmutatás. Ennek tartalmaznia kell leltárszerűen az egyes fájlok pontos nevét, méretét byte-ban, keletkezési idejét, valamint azt, hogy a fájlban mi került megvalósításra.

6.1.1. Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Demo.java	353 byte	2020.03.26 21:05	Demo programosztály

6.1.2. Fordítás

A fenti listában szereplő forrásfájlokból milyen műveletekkel lehet a bináris, futtatható kódot előállítani. Az előállításhoz csak a 2. Követelmények c. dokumentumban leírt környezetet szabad előírni.

```
javac -d bin *.java
```

6.1.3. Futtatás

A futtatható kód elindításával kapcsolatos teendők leírása. Az indításhoz csak a 2. Követelmények c. dokumentumban leírt környezetet szabad előírni.

```
cd bin  
java Main.java
```

6.2. Értékelés

A projekt kezdete óta az értékelésig eltelt időben tagokra bontva, százalékban.

Tag neve	Tag neptun	Munka százalékban
Teszt János	??????	100%

6.3. Napló

A napló tartalmazza az előző beadás óta eltelt időszak történéseit időrendben. A naplóból egyértelműen ki kell derülnie, hogy az egyes anyagrészeket ki és mennyi idő alatt készítette.

A napló bejegyzésekből áll. Minden bejegyzésnek tartalmaznia kell:

- a történes kezdetének időpontját, nap-óra pontossággal
- történes időtartamát, óra felbontással
- a szereplő(k) nevét (Kérjük a szereplők VEZETÉKNEVÉT használni)
- a tevékenység leírását.

Amennyiben a tevékenységben több szereplő vesz részt, akkor aza tevékenység csak értekezlet lehet, amelynek az eredményei DÖNTÉSEK. A döntéseket precízen meg kell szövegezni (Pl.: Az X objektum Y és Z metódusainak kódját W készíti el Q határidőre). Ha a bejegyzés egyetlen személyhez kötődik, akkor meg kell adni, hogy a tevékenység milyen dologra irányul. A dolog a feladat kapcsán elkészítendő termék, amelynek a (esetleg korábban) beadott anyagban megtalálhatónak kell lenni. A naplóbejegyzés felbontásának egysége szöveges, rajzos anyag esetében az ábra, diagram, vagy kb. fél-egy oldalnyi szöveg. Kódban az egység a metódus. (Pl.: A 3. ábrán látható szekvencia-diagram kidolgozása, vagy az X objektum Y és Z metódusainak kódolása és belövése.

Kezdet	Időtartam	Résztvevők	Leírás
feb. 18. 16h	1 óra	Csapat	Értekezlet Döntés: Segédeszközök kiválasztása (git, trello, drive)

7. fejezet

Prototípus koncepciója

A prototípus program célja annak demonstrálása, hogy a program elkészült, helyesen működik, valamennyi feladatát teljesíti. A prototípus változat egy elkészült program kivéve a kifejlett grafikus interfészt. Ez a program is parancssorból futtatható és karakteres ernyőkezelést alkalmaz. Az ütemezés, az aktív objektumok kezelése megoldott. A business objektumok -a megjelenítésre vonatkozó részeket kivéve - valamennyi metódusa a végleges algoritmusokat kell, hogy tartalmazza. A megjelenítés és működtetés egy alfanumerikus képernyőn vezérelhető és követhető, ugyanakkor a vezérlés fájlból is történhet és a megjelenítés fájlba is logolható, ezzel megteremtve a rendszer tesztelésének lehetőségét. Különös figyelmet kell fordítani a parancssori interfész logikájára, felépítésére, valamint arra, hogy az mennyiben tükrözi és teszi láthatóvá a program működését, a beavatkozások hatásait.

Amennyiben változott a modell:

7.0. Változás hatása a modellre

7.0.1. Módosult osztálydiagram

Az analízis modell osztálydiagramja a változások figyelembevételével.

7.0.2. Új vagy megváltozott metódusok

Az analízis modell osztályleírásaiból azon metódusok újbóli felsorolása leírással együtt, amelyek a változtatás miatt módosultak vagy újonnan be lettek vezetve

A bevezetett új osztályok

7.0.2.1. Osztály1

■ Felelősség

Felelősség leírása

■ Ősosztály

Ős-Ősosztály \rightarrow Ősosztály...

■ Metódusok

 $\diamond +B(A\ a) : void$ - metódus B**A korábbi osztályokban történt módosítások****7.0.2.2. Osztály2**

■ Felelősség

Felelősség leírása

■ Ősosztály

Ős-Ősosztály \rightarrow Ősosztály...

■ Metódusok

 $\diamond +B(A\ a) : void$ - metódus B**7.0.3. Szekvencia-diagramok**

Az analízis modell szekvenciadiagramjaiból a változás által érintett, előírt, módosított diagramok



7.1. ábra. Megváltozott szekvenciák

7.1. Prototípus interface-definíciója

Definiálni kell a teszteket leíró nyelvet. Külön figyelmet kell fordítani arra, hogy ha a rendszer véletlen elemeket is tartalmaz, akkor a véletlenszerűség ki-bekapcsolható legyen, és a program determinisztikusan is tesztelhető legyen.

7.1.1. Az interfész általános leírása

A protó (karakters) input és output felületeit úgy kell kialakítani, hogy az input fájlból is vehető legyen illetőleg az output fájlba menthető legyen, vagyis kommunikációra csak a szabványos be- és kimenet használható.

7.1.2. Bemeneti nyelv

Definiálni kell a teszteket leíró nyelvet. Külön figyelmet kell fordítani arra, hogy ha a rendszer véletlen elemeket is tartalmaz, akkor a véletlenszerűség ki-bekapcsolható legyen, és a program determinisztikusan is futtatható legyen. A szálkezelést is tesztelhető, irányítható módon kell megoldani. A programot egy adott konfigurációból is el kell tudni indítani, vagyis kell olyan parancs, amivel konkrét előre megadott állapotból indul a rendszer (pl. load).

- Parancs1
 - Leírás:
 - Opciók:
- Parancs2
 - Leírás:
 - Opciók:

Példák a bemeneti nyelvre:

```
parancs1 arg1 arg2 arg3
parancs2 arg1 arg2
```

Ha szükséges, meg kell adni a konfigurációs (pl. pályaképet megadó) fájlok nyelvtanát is.

7.1.3. Kimeneti nyelv

Egyértelműen definiálni kell, hogy az egyes bemeneti parancsok végrehajtása után előálló állapot milyen formában jelenik meg a szabványos kimeneten. A program képes legyen olyan kimenetet előállítani, amellyel az objektumok állapota ellenőrizhető (pl. save). Ebben az alfejezetben is precízen definiálni kell, hogy a kimenet nyelve milyen elemekből és milyen szintakszissal áll elő.

kimeneti szintaktika

7.2. Összes részletes use-case

Use-case neve:	USE-CASE NEVE
Rövid leírás:	Az eset rövid leírása
Aktorok:	Aktorok
Forgatókönyv:	Forgatókönyv A.1 Alternatíva

7.3. Tesztelési terv

A tesztelési tervben definiálni kell, hogy a be- és kimeneti fájlok egybevetésével miként végezhető el a program tesztelése. Meg kell adni teszt forgatókönyveket. Az egyes tesztek elég informálisan, szabad szöveggént leírni. Teszt-esetenként egy-öt mondatban. Minden teszthez meg kell adni, hogy mi a célja, a proto mely funkcionalitását, osztályait stb. teszteli. Az alábbi táblázat minden teszt-esethez külön-külön elkészítendő.

Teszt-eset neve	TESZTESET NEVE
Rövid leírás	Leírás
Teszt célja	A végrehajtott teszt célja

7.4. Tesztelést támogató segéd- és fordítóprogramok specifikálása

Specifikálni kell a tesztelést támogató segédprogramokat. Rövid bemutatással (elvárt funkcionalitás) specifikálni kell a tesztelést támogató segédprogramokat.

7.5. Napló

A napló tartalmazza az előző beadás óta eltelt időszak történéseit időrendben. A naplóból egyértelműen ki kell derülnie, hogy az egyes anyagrészeket ki és mennyi idő alatt készítette.

A napló bejegyzésekből áll. Minden bejegyzésnek tartalmaznia kell:

- a történes kezdetének időpontját, nap-óra pontossággal
- történes időtartamát, óra felbontással
- a szereplő(k) nevét (Kérjük a szereplők VEZETÉKNEVÉT használni)
- a tevékenység leírását.

Amennyiben a tevékenységben több szereplő vesz részt, akkor aza tevékenység csak értekezlet lehet, amelynek az eredményei DÖNTÉSEK. A döntéseket precízen meg kell szövegezni (Pl.: Az X objektum Y és Z metódusainak kódját W készíti el Q határidőre). Ha a bejegyzés egyetlen személyhez kötődik, akkor meg kell adni, hogy a tevékenység milyen dologra irányul. A dolog a feladat kapcsán elkészítendő termék, amelynek a (esetleg korábban) beadott anyagban megtalálhatónak kell lenni. A naplóbejegyzés felbontásának egysége szöveges, rajzos anyag esetében az ábra, diagram, vagy kb. fél-egy oldalnyi szöveg. Kódban az egység a metódus. (Pl.: A 3. ábrán látható szekvencia-diagram kidolgozása, vagy az X objektum Y és Z metódusainak kódolása és belövése.

Kezdet	Időtartam	Részrtvevők	Leírás
feb. 18. 16h	1 óra	Csapat	Értekezlet Döntés: Segédeszközök kiválasztása (git, trello, drive)

8. fejezet

Részletes tervek

A dokumentum célja, hogy pontosan specifikálja az implementálandó osztályokat, beleértve a privát attribútumokat és metódusokat, ezek definícióját is. A dokumentum második fele részletesen be kell mutassa a korábban definiált be-és kimeneti nyelv szintaxisát felhasználva, hogy mely tesztekkel lesz a prototípus ellenőrizve.

8.1. Osztályok és metódusok tervei

8.1.1. Osztály1

■ Felelősség

Felelősség leírása

■ Interfészek

Megvalósított interfészek felsorolása

■ Ősosztály

Ős-Ősosztály \rightarrow Ősosztály...

■ Attribútumok

$\diamond +A [0..*]$ - adattag A

Milyen publikus, protected és privát metódusokkal rendelkezik. Metódusonként precíz leírás, ha szükséges, activity diagram is a metódusban megvalósítandó algoritmusról.

Minden olyan metódusnak szerepelnie kell, amelyiket az osztály megvalósít vagy felüldefiniál.

■ Metódusok

$\diamond +B(A\ a) : void$ - metódus B

8.1.2. Osztály2

■ Felelősség

Felelősség leírása

■ Állapotgép

A belső működéshez tartozó állapotgép



8.1. ábra. Az osztály állapotváltozásai

8.2. A tesztek részletes tervei, leírásuk a teszt nyelvén

A tesztek részletes tervei alatt meg kell adni azokat a bemeneti adatsorozatokot, amelyekkel a program működése ellenőrizhető. Minden bemenő adatsorozathoz definiálni kell, hogy az adatsorozat végrehajtásától a program mely részeinek, funkcióinak ellenőrzését várjuk és konkrétan milyen eredményekre számítunk, ezek az eredmények hogyan vethetők össze a bemenetekkel. A tesztek leírásakor az előző dokumentumban (proto koncepciója) megadott szintakszist kell használni.

8.2.1. Teszteset1

■ Leírás

Szöveges leírás, kb 1-5 mondat

■ Ellenőrzött funkcionalitás, várható hibahelyek

...

■ Bemenet

A proto nyelvén megadva, pl.:

```
init world
0: S 3
...
print
```

■ Elvárt kimenet

A poro kimeneti nyelvén megadva, pl.:

```
worlddata
...
creatures 0
```


8.3. A tesztelést támogató programok tervei

A tesztadatok előállítására, a tesztek eredményeinek kiértékelésére szolgáló segédprogramok részletes terveit kell elkészíteni.

8.4. Napló

A napló tartalmazza az előző beadás óta eltelt időszak történéseit időrendben. A naplóból egyértelműen ki kell derülnie, hogy az egyes anyagrészeket ki és mennyi idő alatt készítette.

A napló bejegyzésekből áll. Minden bejegyzésnek tartalmaznia kell:

- a történet kezdetének időpontját, nap-óra pontossággal
- történet időtartamát, óra felbontással
- a szereplő(k) nevét (Kérjük a szereplők VEZETÉKNEVÉT használni)
- a tevékenység leírását.

Amennyiben a tevékenységben több szereplő vesz részt, akkor a tevékenység csak értekezlet lehet, amelynek az eredményei DÖNTÉSEK. A döntéseket precízen meg kell szövegezni (Pl.: Az X objektum Y és Z metódusainak kódját W készíti el Q határidőre). Ha a bejegyzés egyetlen személyhez kötődik, akkor meg kell adni, hogy a tevékenység milyen dologra irányul. A dolog a feladat kapcsán elkészítendő termék, amelynek a (esetleg korábban) beadott anyagban megtalálhatónak kell lenni. A naplóbejegyzés felbontásának egysége szöveges, rajzos anyag esetében az ábra, diagram, vagy kb. fél-egy oldalnyi szöveg. Kódban az egység a metódus. (Pl.: A 3. ábrán látható szekvencia-diagram kidolgozása, vagy az X objektum Y és Z metódusainak kódolása és belövése.

Kezdet	Időtartam	Részrtvevők	Leírás
feb. 18. 16h	1 óra	Csapat	Értekezlet Döntés: Segédeszközök kiválasztása (git, trello, drive)

10. fejezet

Prototípus beadása

10.1. Fordítási és futtatási útmutató

A feltöltött program fordításával és futtatásával kapcsolatos útmutatás. Ennek tartalmaznia kell leltárszerűen az egyes fájlok pontos nevét, méretét byte-ban, keletkezési idejét, valamint azt, hogy a fájlban mi került megvalósításra.

10.1.1. Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Demo.java	353 byte	2020.03.26 21:05	Demo programosztály

10.1.2. Fordítás

A fenti listában szereplő forrásfájlokból milyen műveletekkel lehet a bináris, futtatható kódot előállítani. Az előállításhoz csak a 2. Követelmények c. dokumentumban leírt környezetet szabad előírni.

```
javac -d bin *.java
```

10.1.3. Futtatás

A futtatható kód elindításával kapcsolatos teendők leírása. Az indításhoz csak a 2. Követelmények c. dokumentumban leírt környezetet szabad előírni.

```
cd bin
java Main.java
```

10.2. Tesztek jegyzőkönyvei

10.2.1. Teszteset1

Az alábbi táblázatot a megismételt (hibás) tesztek esetén kell kitölteni minden ismétléshez egyszer. Ha szükséges, akkor a valós kimenet is mellékelhető mint a teszt eredménye

Tesztelő neve	????
Teszt időpontja	Ápr. 25. 13:50
Teszt eredménye	A teszt eredménye
Lehetséges hibaokok	A teszt hibás futásának okai
Változtatások	Változtatások

Az alábbi táblázatot az utolsó, sikeres tesztfuttatáshoz kell kitölteni

Tesztelő neve	????
Teszt időpontja	Ápr. 25. 14:00

10.3. Értékelés

A projekt kezdete óta az értékelésig eltelt időben tagokra bontva, százalékban.

Tag neve	Tag neptun	Munka százalékban
Teszt János	??????	100%

10.4. Napló

A napló tartalmazza az előző beadás óta eltelt időszak történéseit időrendben. A naplóból egyértelműen ki kell derülnie, hogy az egyes anyagrészeket ki és mennyi idő alatt készítette.

A napló bejegyzésekből áll. Minden bejegyzésnek tartalmaznia kell:

- a történes kezdetének időpontját, nap-óra pontossággal
- történes időtartamát, óra felbontással
- a szereplő(k) nevét (Kérjük a szereplők VEZETÉKNEVÉT használni)
- a tevékenység leírását.

Amennyiben a tevékenységben több szereplő vesz részt, akkor aza tevékenység csak értekezlet lehet, amelynek az eredményei DÖNTÉSEK. A döntéseket precízen meg kell szövegezni (Pl.: Az X objektum Y és Z metódusainak kódját W készíti el Q határidőre). Ha a bejegyzés egyetlen személyhez kötődik, akkor meg kell adni, hogy a tevékenység milyen dologra irányul. A dolog a feladat kapcsán elkészítendő termék, amelynek a (esetleg korábban) beadott anyagban megtalálhatónak kell lenni. A naplóbejegyzés felbontásának egysége szöveges, rajzos anyag esetében az ábra, diagram, vagy kb. fél-egy oldalnyi szöveg. Kódban az egység a metódus. (Pl.: A 3. ábrán látható szekvencia-diagram kidolgozása, vagy az X objektum Y és Z metódusainak kódolása és belövése.

Kezdet	Időtartam	Résztvevők	Leírás
feb. 18. 16h	1 óra	Csapat	Értekezlet Döntés: Segédeszközök kiválasztása (git, trello, drive)

11. fejezet

Grafikus felület specifikációja

11.1. A grafikus interfész

A menürendszer, a kezelői felület grafikus képe. A grafikus felület megjelenését, a használt ikonokat, stb screenshot-szerű képekkel kell bemutatni. Az építészetben ez a homlokzati terv.



11.1. ábra. Példa kép

11.2. A grafikus rendszer architektúrája

A felület működésének elve, a grafikus rendszer architektúrája (struktúra diagramok). A struktúra diagramokon a prototípus azon és csak azon osztályainak is szerepelnie kell, amelyekhez a grafikus felületet létrehozó osztályok kapcsolódnak.

11.2.1. A felület működési elve

Le kell írni, hogy a grafikai megjelenésért felelős osztályok, objektumok hogyan kapcsolódnak a meglevő rendszerhez, a megjelenítés során mi volt az alapelv. Törekedni kell az MVC megvalósításra. Alapelvek lehetnek: **push** alapú: a modell értesíti a felületet, hogy változott; **pull** alapú: a felület kérdezi le a modellt, hogy változott-e; **kevert**: a kettő kombinációja.

11.2.2. A felület osztály-struktúrája

Osztálydiagram. Minden új osztály, és azon régiak, akik az újakhoz közvetlenül kapcsolódnak.



11.2. ábra. Példa kép

11.3. A grafikus objektumok felsorolása

Az új osztályok felsorolása. Az régi osztályok közül azoknak a felsorolása, ahol változás volt. Ezek esetén csak a változásokat kell leírni.

11.3.1. Új osztályok

11.3.1.1. Osztály1

- Felelősség

Felelősség leírása

- Interfészek

Megvalósított interfészek felsorolása

- Ősosztály

Ős-Ősosztály → Ősosztály...

- Attribútumok

◇ $+A [0..*]$ - adattag A

Milyen publikus, protected és privát metódusokkal rendelkezik. Metódusonként precíz leírás, ha szükséges, activity diagram is a metódusban megvalósítandó algoritmusról.

Minden olyan metódusnak szerepelnie kell, amelyiket az osztály megvalósít vagy felüldefiniál.

- Metódusok

◇ $+B(A\ a) : void$ - metódus B

11.3.2. Megváltozott osztályok

11.3.2.1. Osztály2

- Felelősség

Felelősség leírása

- Interfészek

Megvalósított interfészek felsorolása

■ Ősosztály

Ős-Ősosztály \rightarrow Ősosztály...

■ Attribútumok

$\diamond +A [0..*]$ - adattag A

Milyen publikus, protected és privát metódusokkal rendelkezik. Metódusonként precíz leírás, ha szükséges, activity diagram is a metódusban megvalósítandó algoritmusról.

Minden olyan metódusnak szerepelnie kell, amelyiket az osztály megvalósít vagy felüldefiniál.

■ Metódusok

$\diamond +B(A\ a) : void$ - metódus B

11.4. Kapcsolat az alkalmazói rendszerrel

Szekvencia-diagramokon ábrázolni kell a grafikus rendszer működését. Konzisztens kell legyen az előző alfejezetekkel. Minden metódus, ami ott szerepel, fel kell tűnjön valamelyik szekvenciában. Minden metódusnak, ami szekvenciában szerepel, szereplnie kell a valamelyik osztálydiagramon.

11.5. Napló

A napló tartalmazza az előző beadás óta eltelt időszak történéseit időrendben. A naplóból egyértelműen ki kell derülnie, hogy az egyes anyagrészeket ki és mennyi idő alatt készítette.

A napló bejegyzésekből áll. Minden bejegyzésnek tartalmaznia kell:

- a történes kezdetének időpontját, nap-óra pontossággal
- történes időtartamát, óra felbontással
- a szereplő(k) nevét (Kérjük a szereplők VEZETÉKNEVÉT használni)
- a tevékenység leírását.

Amennyiben a tevékenységben több szereplő vesz részt, akkor aza tevékenység csak értekezlet lehet, amelynek az eredményei DÖNTÉSEK. A döntéseket precízen meg kell szövegezni (Pl.: Az X objektum Y és Z metódusainak kódját W készíti el Q határidőre). Ha a bejegyzés egyetlen személyhez kötődik, akkor meg kell adni, hogy a tevékenység milyen dologra irányul. A dolog a feladat kapcsán elkészítendő termék, amelynek a (esetleg korábban) beadott anyagban megtalálhatónak kell lenni. A naplóbejegyzés felbontásának egysége szöveges, rajzos anyag esetében az ábra, diagram, vagy kb. fél-egy oldalnyi szöveg. Kódban az egység a metódus. (Pl.: A 3. ábrán látható szekvencia-diagram kidolgozása, vagy az X objektum Y és Z metódusainak kódolása és belövése.

Kezdet	Időtartam	Részrtvevők	Leírás
feb. 18. 16h	1 óra	Csapat	Értekezlet Döntés: Segédeszközök kiválasztása (git, trello, drive)

13. fejezet

Grafikus felület specifikációja

13.1. Fordítási és futtatási útmutató

A feltöltött program fordításával és futtatásával kapcsolatos útmutatás. Ennek tartalmaznia kell leltárszerűen az egyes fájlok pontos nevét, méretét byte-ban, keletkezési idejét, valamint azt, hogy a fájlban mi került megvalósításra.

13.1.1. Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Demo.java	353 byte	2020.03.26 21:05	Demo programosztály

13.1.2. Fordítás

A fenti listában szereplő forrásfájlokból milyen műveletekkel lehet a bináris, futtatható kódot előállítani. Az előállításhoz csak a 2. Követelmények c. dokumentumban leírt környezetet szabad előírni.

```
javac -d bin *.java
```

13.1.3. Futtatás

A futtatható kód elindításával kapcsolatos teendők leírása. Az indításhoz csak a 2. Követelmények c. dokumentumban leírt környezetet szabad előírni.

```
cd bin  
java Main.java
```

13.2. Értékelés

A projekt kezdete óta az értékelésig eltelt időben tagokra bontva, százalékban.

Tag neve	Tag neptun	Munka százalékban
Teszt János	??????	100%

13.3. Napló

A napló tartalmazza az előző beadás óta eltelt időszak történéseit időrendben. A naplóból egyértelműen ki kell derülnie, hogy az egyes anyagrészeket ki és mennyi idő alatt készítette.

A napló bejegyzésekből áll. Minden bejegyzésnek tartalmaznia kell:

- a történes kezdetének időpontját, nap-óra pontossággal
- történes időtartamát, óra felbontással
- a szereplő(k) nevét (Kérjük a szereplők VEZETÉKNEVÉT használni)
- a tevékenység leírását.

Amennyiben a tevékenységben több szereplő vesz részt, akkor aza tevékenység csak értekezlet lehet, amelynek az eredményei DÖNTÉSEK. A döntéseket precízen meg kell szövegezni (Pl.: Az X objektum Y és Z metódusainak kódját W készíti el Q határidőre). Ha a bejegyzés egyetlen személyhez kötődik, akkor meg kell adni, hogy a tevékenység milyen dologra irányul. A dolog a feladat kapcsán elkészítendő termék, amelynek a (esetleg korábban) beadott anyagban megtalálhatónak kell lenni. A naplóbejegyzés felbontásának egysége szöveges, rajzos anyag esetében az ábra, diagram, vagy kb. fél-egy oldalnyi szöveg. Kódban az egység a metódus. (Pl.: A 3. ábrán látható szekvencia-diagram kidolgozása, vagy az X objektum Y és Z metódusainak kódolása és belövése.

Kezdet	Időtartam	Részrtvevők	Leírás
feb. 18. 16h	1 óra	Csapat	Értekezlet Döntés: Segédeszközök kiválasztása (git, trello, drive)

14. fejezet

Összefoglalás

14.1. A projektre fordított összes munkaidő

Személyenként és összesen

Tag neve	Tag neptun	Munka órában
TAG1	NEPTUN1	ÓRA1
TAG2	NEPTUN2	ÓRA2
TAG3	NEPTUN3	ÓRA3
TAG4	NEPTUN4	ÓRA4
TAG5	NEPTUN5	ÓRA5
Csapatgyűlés	-	-
Összesen	-	Σ

14.2. A feltöltött programok forrássorainak megoszlása

Fázis	Kódsorok száma
Szkeleton	??
Prototípus	??
Grafikus változat	??
Összesen	??

14.3. Projekt összegzés

A projekt tapasztalatait összegző részben a csapatoknak a projektről kialakult véleményét várjuk. A megválaszolandók köre az alábbi

- *Mit tanultak a projektből konkrétan és általában?*

...

- *Mi volt a legnehezebb és a legkönnyebb?*
...
- *Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal?*
...
- *Ha nem, akkor hol okozott ez nehézséget?*
...
- *Milyen változtatási javaslatuk van?*
...
- *Milyen feladatot ajánlanának a projektre?*
...