

Objektumorientált szoftverfejlesztés

Vizsgalap

Tétel sorszáma: 07

Értékelés:

elégtelen (1)

elégséges (2)

közepes (3)

jó (4)

jeles (5)

Vizsgázó

Neve: Tomor Péter

Neptun kódja: GA1HPQ

Vizsga dátuma: 2019 01 11 14:30

Kérdés

Az objektumok közötti kapcsolatokat mikor jelöljük ismeretségeknek és mikor tartalmazásnak? Mi a különbség az aggregáció és a kompozíció között?

Válasz

Az objektumok közötti kapcsolatokat közül az ismertségi kapcsolatnak jelöljük, amikor az objektumok egymástól függetlenül léteznek: pl szülő-gyermek kapcsolat. A szülő nem tartalmazza a gyermeket (a terhességi fázistól tekintünk el), vagy például a házastársak közötti kapcsolat.

Tartalmazásnak jelöljük a kapcsolatot amikor az egyik objektum része a másiknak. A tartalmazó az összetett objektum a benne lévő pedig a részobjektum. pl: Doboz - bombon kapcsolat vagy a terem-asztal kapcsolat.

Az aggregáció: egész-rész kapcsolat egy fajtája "gyenge tartalmazásnak" is nevezzük, pl terem-asztal kapcsolat. Az egész létezhet a része nélkül is.

Kompozíció: az egész-rész kapcsolat másik fajtája "erős tartalmazásnak" is nevezzük pl: Háztető-gerenda kapcsolat. Az egész nem létezik a része nélkül. Az egész élettartama megegyezik a részével.

A különbség a kettő között pontosan abban található, hogy az egész létezhet-e a része nélkül, kompozíciónál nem.

Az ismeretségi és tartalmazási kapcsolat nem „helyraji”, hanem függési kérdés. Két objektum között akkor értelmezünk tartalmazási kapcsolatot, ha létük összefügg – függetlenül attól, hogy fizikailag egymáshoz képest hol helyezkednek el (pl. a memóriában). A kapcsolat típusának ilyen vagy olyan beállítása azt közli az implementációval, hogy az egyes objektumoknak hogyan kell viselkedniük egymással kapcsolatban – ez a kódolási megoldás számára ad iránymutatást.

Valamikor egyébként valóban volt egy fizikai értelmezése is a tartalmazási kapcsolatnak, amikor a memóriában a „tartalmazott” objektum valóban a „tartalmazó” adatterületén belül helyezkedett el – ez ma már nem releváns.

Az aggregáció és kompozíció definíciója helyes, de láthatólag nem érti tervezési alkalmazásuk célját. Ezt mutatja a szülő-gyerek kapcsolat példa is: a gyerek igenis függ a szüleitől, sokszor még születése utáni létében is – fordítva ez már nem igaz. Tehát kapcsolatuk aggregáció típusú tartalmazás. A kompozícióra adott példája szintén helytelen: ha igaz lenne amit ír, nem lehetne tetőt építeni, mert tető nélkül nem lehetne gerenda, de ha nincs gerenda, hogyan építünk tetőt?

Feladat

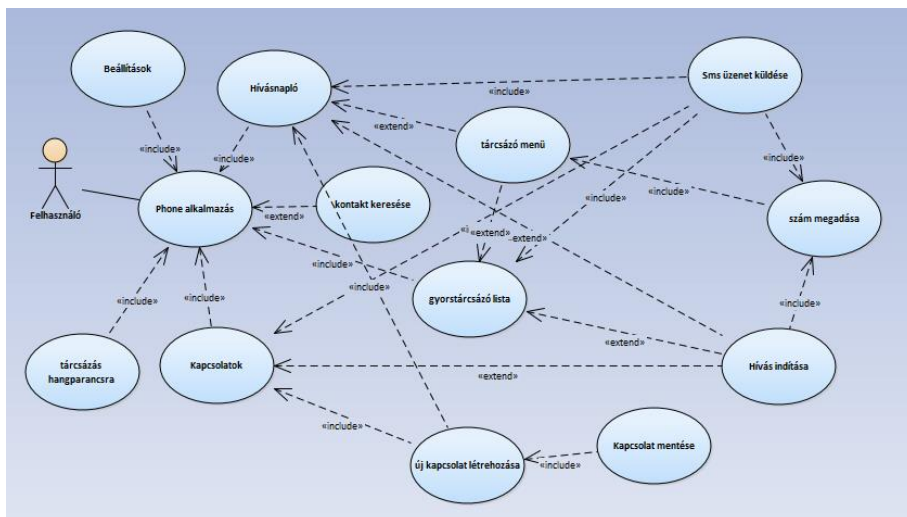
Mobilhívás usecase modell

Egy mobiltelefon sokféle szolgáltatást nyújt a használójának. Ezek közül az egyik alapvető a hívások kezdeményezése. Telefonján térképezze fel a hívás funkciót és a hozzá kapcsolódó egyéb szolgáltatásokat, és ábrázolja ezeket használati eset modellben. Az egyes usecase-ek között alkalmazott include, extend vagy öröklési kapcsolatok választását indokolja meg.

A modellnek adjon címet, amiben feltünteti a telefon gyártmányát és operációs rendszerét + a következő szöveget: "Hívás usecase modell".

Megoldás

A ZTE Axon 7, Android 8.0 Hívás usecase modellje:



A Phone alkalmazásból közvetlenül elérhető a beállítások, a hívásnapló, tárcsázás hangparancsra, kapcsolatok, és a gyors tárcsázó. A legtöbb menüből elérhető a hívás és az sms küldés funkció. új kapcsolatot lehet létrehozni hívásnaplóból és kapcsolatok menüből is.

A használati esetek egy része helyes, más része nem ide tartozik vagy nem használati eset. A közöttük levő kapcsolatok helytelenek – nem érti ezek funkcióját:

- Include: két szolgáltatás kötelező jellegű összekapcsolása. A nyíl forrása kötelezően beépíti működésébe a megcélzott funkciót.
- Extend: két szolgáltatás opcionális összekapcsolása. A nyíl forrása feltételek teljesülése vagy a felhasználó választása nyomán „kiegészíti” a megcélzott funkciót.

Lásd online tananyag: Modellek készítése/Követelmény modellek/Funkcionális követelmények/Feladatmodell/, A usecase modell elemei

formázott: Listaszerű bekezdés, Felsorolás + Szint: 1 + Igazítás: 0,63 cm + Behúzás: 1,27 cm

formázott: Betűtípus: 9 pt

formázott: Betűtípus: 12 pt

formázott: Betűtípus: (Alapérték) +Szövegtörzs (Calibri), 9 pt, Nem Félkövér

A megoldás során szem előtt tartandók:

1. A feladat megoldásához először is pontosan rögzíteni kell a határokat. A feladatleírás a telefonhívás témakörére koncentrá – minden más nem tartozik ide, legyen akár „elérhető” bármelyik, egyébként a hívással kapcsolatos felületről.

- a. a beállítások,
- b. az SMS küldés nem tartoznak ide.

2. Egy alkalmazás menüpontjai nem (mindig) egyenlőek a használati eseteivel.

- a. A kapcsolat mentése a kapcsolat létrehozás / szerkesztés usecase része, ami a feladatunk látószögéből nézve annak egy belső művelete – ebben a modellben „nem látszik”.

3. A usecase egy funkció, szolgáltatás – tehát

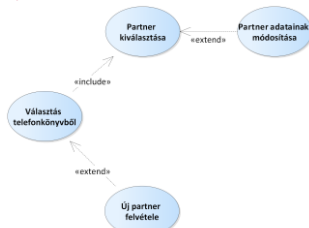
- a. a hívásnapló,
- b. a Phone alkalmazás (amúgy ezen „belül” vagyunk: ennek a használati eseteit kutatjuk).
- c. a kapcsolatok,
- d. a tárcsázó menü,
- e. a gyorstárcsázó lista nem használati esetek.

4. A létező rendszerben előre látott szolgáltatásokat elemezni kell, esetleg fogalmakat kell alkotni és azok szerint rendszerezni a tapasztaltakat.

- a. A szám megadása a hívás egy általános, alapvető funkciója. Ennek vannak különböző válfajai, mint pl. kézi beírás, bemondás, megadás a telefonkönyvből, választás előzményekből. Ezek modellje pl.:



- b. A választás telefonkönyvből történhet létező kapcsolat kiválasztása vagy új felvétele után. A partnert mindenképpen ki kell választani (include), ez előtt esetleg (extend) újként felvesszük.



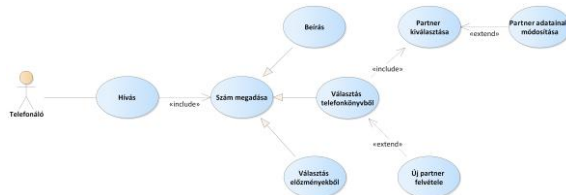
(A kiválasztott partnert esetleg (extend) módosíthatjuk is)

formázott: Behúzás: Bal: 0,63 cm

formázott

c. A híváshoz tehát a számot valamilyen módon kötelezően produkálni kell – aztán jöhet a gombnyomás (a Hívás egy belső művelete).

Tehát a telefonhíváshoz tartozó használati esetek:



A fentiek kiegészíthetők az Oreó-ban esetleg megvalósított új, ide tartozó hívásfunkciókkal. Én nem nagyon láttam ilyeneket.