

Szoftverfejlesztési technológiák 10-12

/óravázlat/

Szekvenciadiagram

A rendszer *működésének* leírására szolgál a szekvenciadiagram. A probléma megoldása során az objektumok egymásnak üzeneteket küldenek. Az üzenetek időbeli sorrendje szemléltetésének egyik eszköze a szekvenciadiagram.

A szekvenciadiagram alapfogalmai, komponensei a következők:

- osztályszerep,
- osztályszerep életvonala,
- aktivációs életvonal,
- üzenet.

Osztályszerep

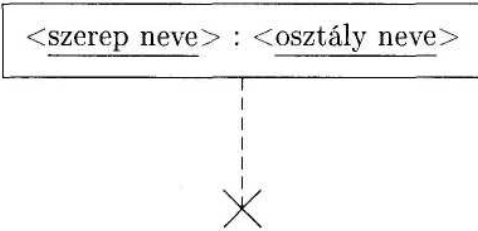
Az osztály szerepét az osztályok közötti üzenetben megtestesítheti az osztály egy vagy több objektuma, amelyek az üzenetküldés szempontjából konform, azaz azonos módon járnak el. Az osztályszerep jelölése:



<szerep neve> : <osztály neve>

Osztályszerep életvonala

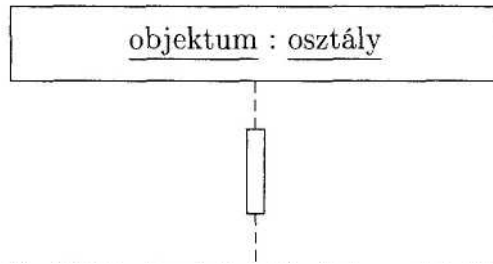
Az életvonal az osztályszerep időben való létezését jelenti. Az osztályszerep életvonalának jelölése a következő, ahol x az osztályszerepet megtestesítő objektum megszűnését jelenti:



<szerep neve> : <osztály neve>

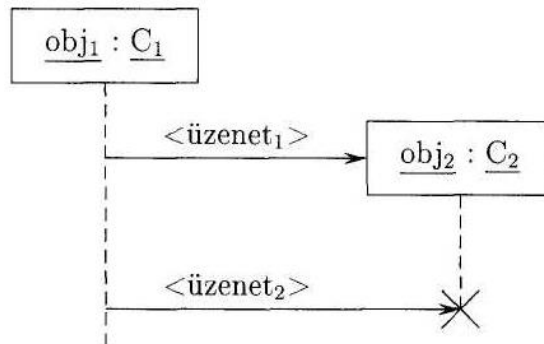
Az osztályszerep aktivációs életvonala

Az aktivációs életvonal az osztályszerepnek azt az állapotát jelöli, amelyben az osztályszerep megtestesítői műveletet hajtanak végre, miközben más objektumok vezérlése alatt állnak. Az aktivációs életvonal jelölése:



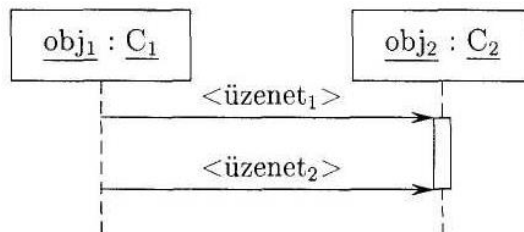
Az objektum létrehozása és megsemmisítése

Egy objektum létrejöhet egy másik objektum létrehozó üzenetének a hatására, és megsemmisülhet, ha a másik objektum egy törlést (lerombolást) jelentő üzenetet ad ki rá. Jelölése:



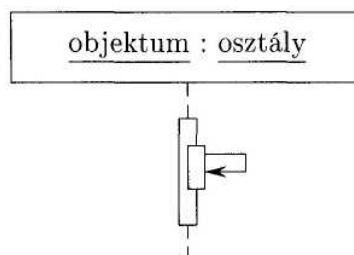
Az objektum aktivációja

Életvonala során az objektum aktív módon viselkedhet, amíg valamilyen tevékenységet végrehajt. Az aktív állapotot előidézhetheti egy másik objektum üzenete, illetve üzenettel meg is szüntetheti azt:

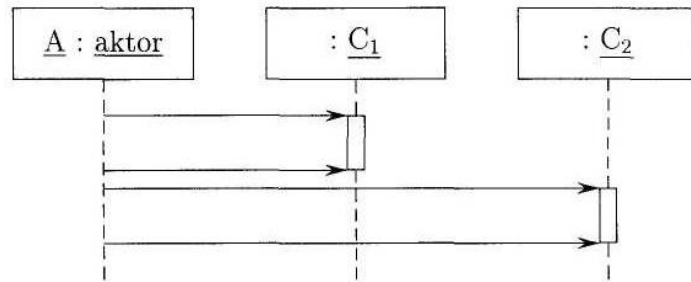


Ha szükséges, az aktív szakasz mellett feltüntethető a tevékenység neve.

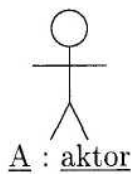
Az objektum aktivációjának egy speciális esete a *rekurzív aktiváció*, amikor egy objektum saját magát aktiválja:



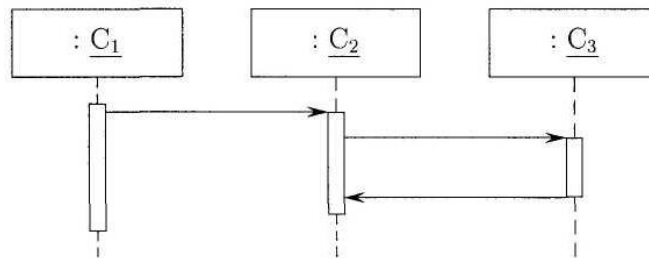
Az aktiváció lehet *centrálisan vezérelt*. Centrális esetben minden objektumot egy *aktor* objektum aktivizál:



Az aktor egy másik szokásos jelölése:



Az aktiváció lehet decentralizáltan vezérelt:

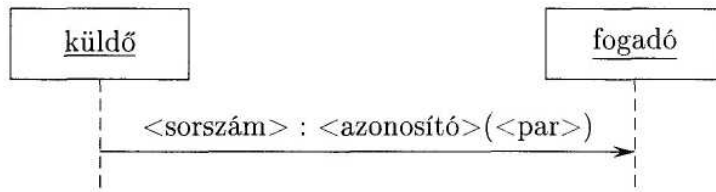


Üzenettípusok

Az eddigiekben láttuk, hogy az objektumok aktivációs életvonalának alakulását az üzenetek szabályozzák. A következőkben ezen üzeneteket osztályozzuk annak alapján, hogy miként viszonyulnak az objektumokhoz. Az üzenet az objektumok közötti információátadás formája. Az üzenet átadása általában egy esemény bekövetkezését takarja. Az üzenet küldésének az a célja, hogy az objektum működésbe hozza a másik objektumot. Az üzenet azok között az objektumok között jöhet létre, amelyek az objektumdiagramban kapcsolatban állnak. Az *üzenet az esemény egy példánya*. Az üzenet küldése egy olyan *akció*, amelynek eredménye egy végrehajtható utasítás. Az üzenetnek van azonosítója (neve, szövege), lehet paramétere, sorszám, stb.

Egyszerű üzenet

Egy aktív objektum üzenetet küld egy passzív objektumnak. Az aktív objektum átadja a vezérlést a passzív objektumnak. Ez tulajdonképpen egy közöségi eljárás:



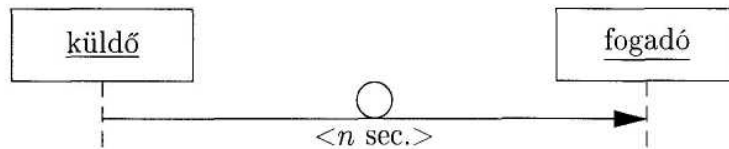
Szinkronizációs üzenet

A küldő objektum elküldi az üzenetet, és a küldő blokkolt állapotba kerül, amíg a fogadó nem fogadta az üzenetet. A küldő várakozik, amíg a szinkronizációs feltétel nem teljesül:



Időhöz kötött várakozás

A küldő legfeljebb a megjelölt ideig várakozik arra, hogy a fogadó fogadja az üzenetet. Ha ennyi idő alatt ez nem következik be, akkor folytatja a tevékenységét:



Randevú üzenet

A fogadó várakozik arra, hogy a küldő üzenetet küldjön neki. A fogadó előbb várakozó állapotba helyezi magát a fogadáshoz:



Aszinkron üzenet

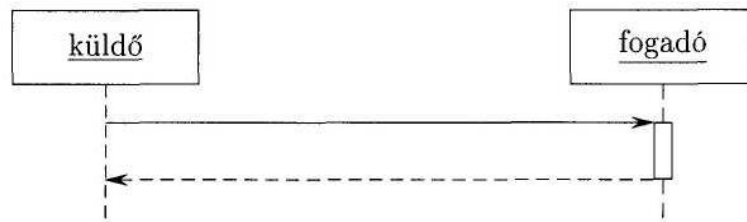
A küldő folyamat nem szakad meg, nem érdekli őt, hogy mikor kapta meg a fogadó az üzenetet:



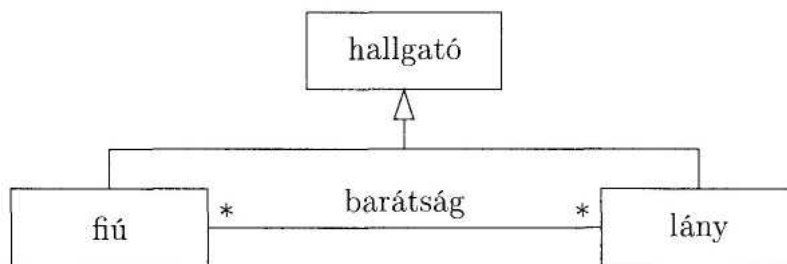
Visszatérési üzenet

Az üzenetet egy aktivizált objektum küldi az aktivizáló objektumnak akkor, amikor befejezi a tevékenységet, és a vezérlést visszaadja. Ezt az üzenetet gyakran nem tüntetjük fel a

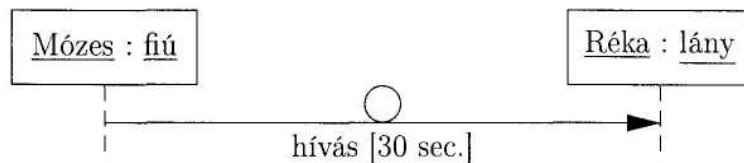
szekvenciadiagramban, mert rendszerint kiolvasható a többi üzenetből és az aktív szakaszokból. Vannak azonban olyan esetek, amikor ennek feltüntetése célszerű:



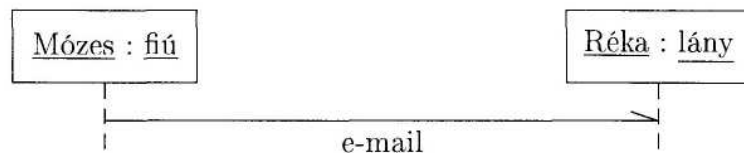
A definíciók után nézzünk néhány példát az egyes üzenetekre. Az első két példában a hallgatók osztályát használjuk, amelyből specializációval kapjuk a fiú és lány hallgatókat, akik barátságban lehetnek egymással:



Időhöz kötött várakoztatásra példa, amikor Mózes telefonon felhívja Rékát, és 30 másodperc vár arra, hogy Réka felvegye a telefont:



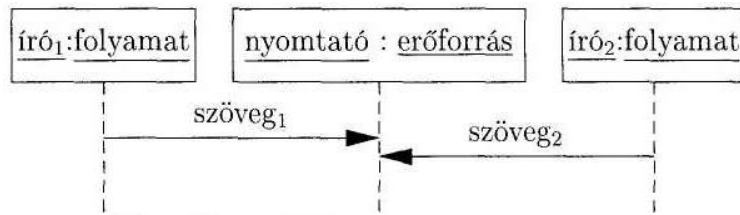
Aszinkron üzenetre egy példa, ha Mózes e-mail-t küld Rékának, majd tanul tovább:



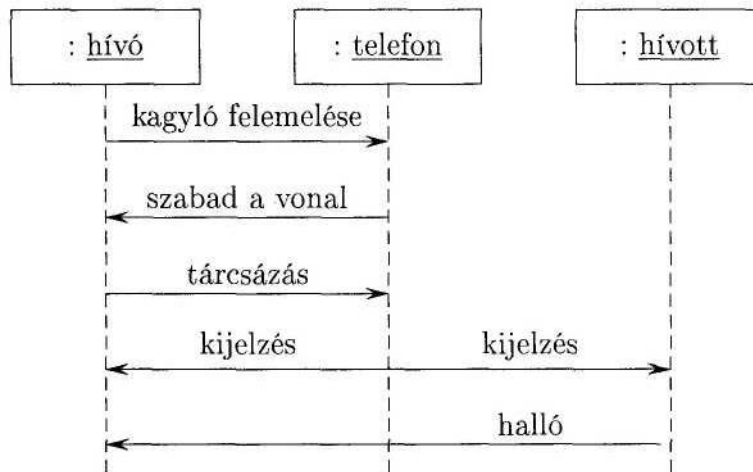
A szinkronizációs üzenetre bemutatandó példához tekintsünk egy kölcsönös kizárásos modellt:



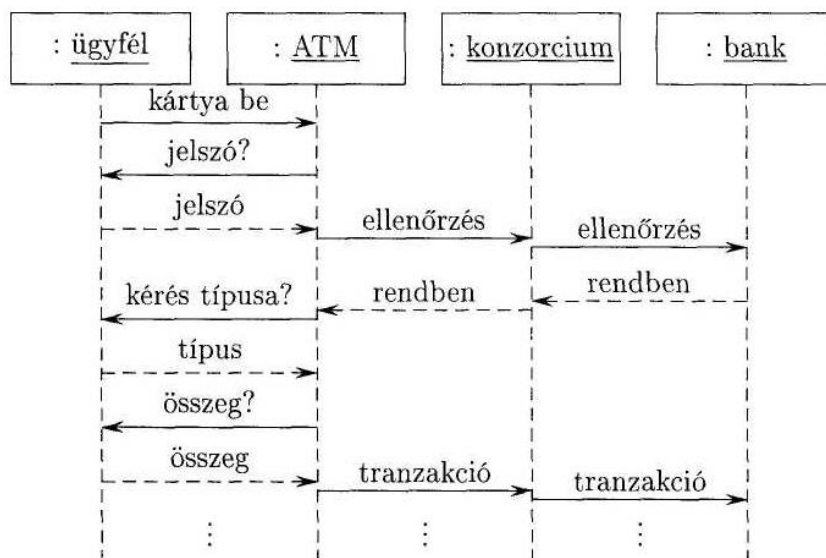
Ennek a modellnek az alapján, ha ketten kívánják használni például a nyomtatót – ami az erőforrás szerepét tölti be – a szöveg kiírására, akkor a kialakult helyzet szekvenciadiagramja:



Példa a decentralizált vezérlésre a szokásos telefonkapcsolat szekvenciadiagramja, amely egyben szemlélteti az objektumok közötti üzenetek időbeli lejátszódását is:



A következő példa legyen az ATM készülék működését szemléltető szekvenciadiagram, amikor szintén csak az üzenetek időben lejátszódó sorrendjét kívánjuk szemléltetni. Ez egy újabb példa a decentralizáltan vezérelt rendszerre:



A szekvenciadiagram kiegészítései

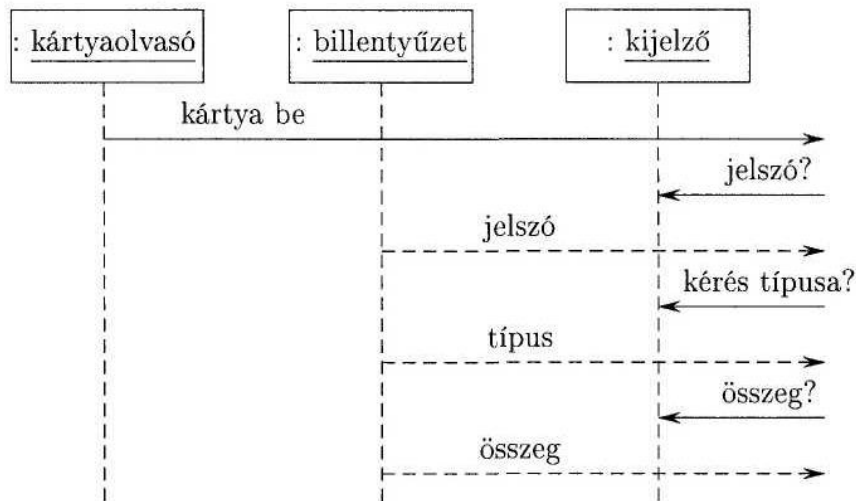
Az eddig bevezetett jelölések felhasználásával viszonylag egyszerű feladatok esetén is nagy méretű, bonyolult és nehezen áttekinthető szekvenciadiagramokhoz jutunk. Ezért célszerűnek látszik az eddigi jelöléseket kiegészíteni, hogy az áttekinthetőséget növeljük. (Az UML jelenlegi változatában erre nincs szabványos módszer, legfeljebb utalások.) Ezek a kiegészítések lehetővé teszik a hierarchikus szerkezet, a választás, az iteráció és a párhuzamosság ábrázolását.

A hierarchia ábrázolása

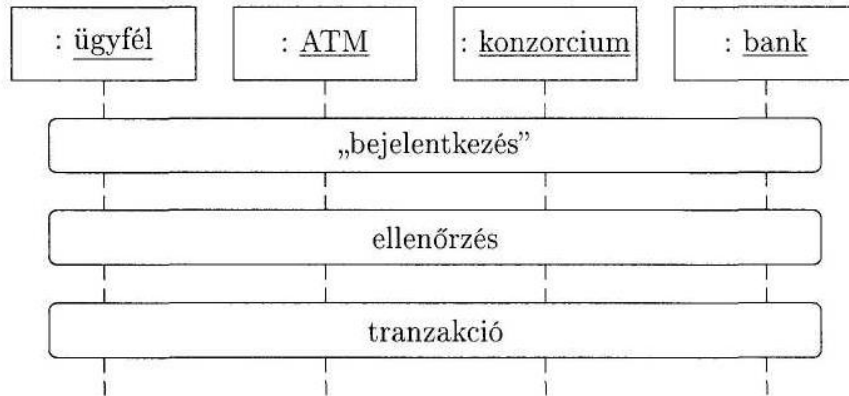
Két módszerrel tudunk új szerkezeti (absztrakciós) szinteket bevezetni egy szekvenciadiagramba: példány dekompozícióval és hivatkozással.

Példány dekompozíció esetén a szekvenciadiagram egyik objektumát, illetve osztályszerepét egy másik szekvenciadiagramban fejtjük ki részletesen. A részletezés során az objektum vagy szerep helyét több objektum vagy szerep veszi át, és pontosítjuk, hogy melyik üzenetet melyik részobjektum küldi, illetve fogadja. Lényeges megszorítás, hogy az üzenetek sorrendje a dekompozíció során nem változhat meg.

Például az ATM készülék működését leíró előbbi szekvenciadiagramban az ügyfél objektum igény esetén tovább bontható kártyaolvasó, billentyűzet és kijelző objektumokra:

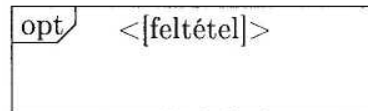


Hivatkozás esetén a szekvenciadiagram egy időintervallumnak megfelelő részét kiemeljük egy külön diagramba. Az időintervallumnak megfelelő rész rendszerint egy (nevesíthető) tevékenységnek felel meg. A hivatkozást egy lekerekített sarkú téglalappal jelöljük, ami egy névvel (azonosítóval) rendelkezik. A kifejtést tartalmazó szekvenciadiagramot ezzel a névvel látjuk el. Az ATM készüléknél maradvá szekvenciadiagramjában például három részt különíthetünk el: „bejelentkezés”, ellenőrzés és tranzakció: (A részeknek megfelelő szekvenciadiagramokat adódnak az eredeti diagramból.)

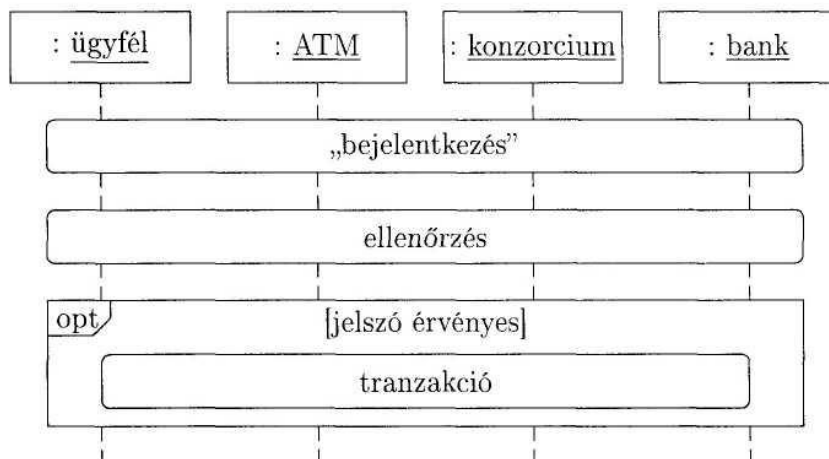


A választások ábrázolása

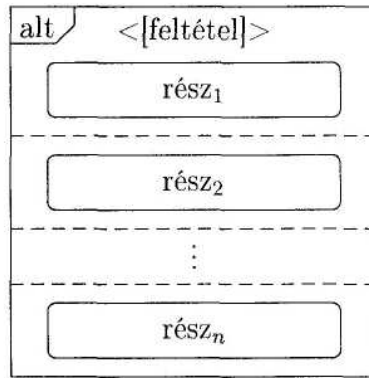
Egy szekvenciadiagramon belül lehetséges, hogy bizonyos rész üzeneteire csak akkor kerül sor, ha valamilyen feltétel teljesül. Ezt nevezzük opcionális résznek. A részt egy téglalapba zárjuk, a téglalap bal felső sarkába az *opt* szót írjuk, és a téglalap felső részének közepére írjuk a feltételt:



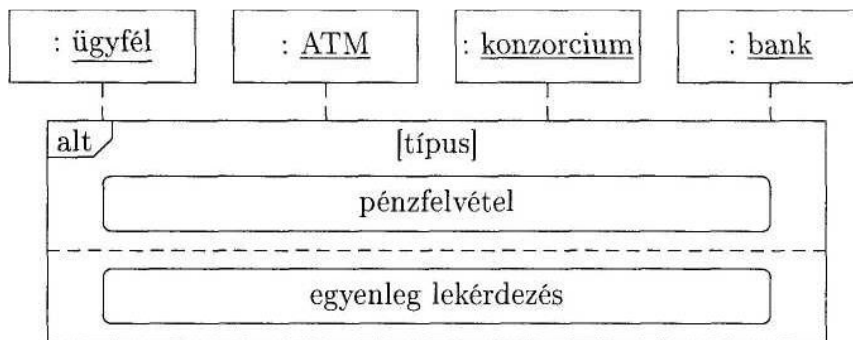
Például az ATM esetében csak akkor kell tranzakciót végrehajtani, ha az ellenőrzés után tudjuk, hogy a jelszó érvényes:



Az is előfordulhat, hogy egy kifejezés értékétől függően eltérő részek lehetnek a szekvenciadiagramban. Ezt nevezzük alternatív résznek. Ezt a részt is téglalapba zárjuk, a bal felső sarokba az *alt* szó kerül, a kifejezést a téglalap felső részének közepére írjuk, az egyes részeket pedig szaggatott vonallal választjuk el egymástól:

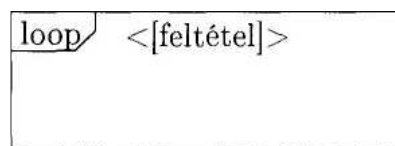


Az ATM esetében mást kell tennünk akkor, ha a kiválasztott művelet a pénzfelvétel, illetve az egyenleg lekérdezése:

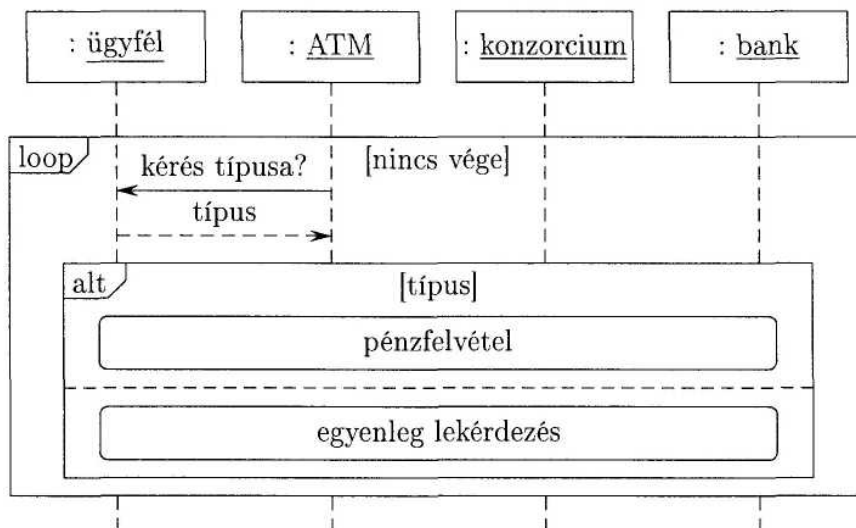


Az iteráció ábrázolása

Egy rendszer működése során bizonyos tevékenységek ciklikusan ismétlődhetnek. Ennek megfelelően a szekvenciadiagram egyes részeinek üzenetei is többször fordulhatnak elő. Az ilyen részt nevezzük iterációs résznek, ciklusnak. A szekvenciadiagramban ezt a részt egy téglalapba foglaljuk, a bal felső sarokba a *loop* szót írjuk, a ciklus feltételét pedig a téglalap felső részének közepére helyezzük:

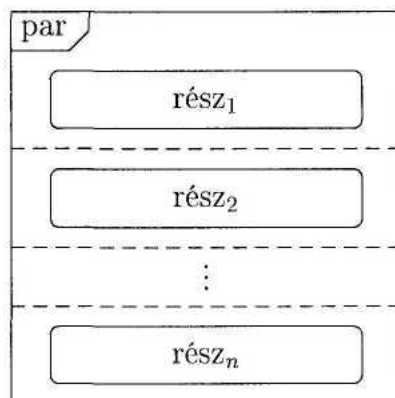


Ha az ATM készülék példájában figyelembe vesszük, hogy több műveletet is végre lehet hajtani egy sikeres bejelentkezés után, akkor a tranzakciós rész egy ciklus lesz, amelyben szerepel az előzőekben megismert alternatív rész:

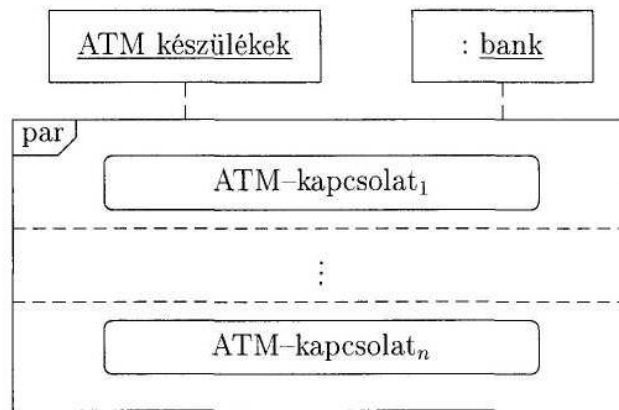


A párhuzamosság ábrázolása

Párhuzamos, osztott rendszerek esetén bizonyos folyamatok egy időben zajlanak. Ezekben az esetekben a szekvenciadiagramban is biztosítanunk kell a párhuzamosság megjelenését. Az egymással egy időben zajló részeket egy téglalapon belül szaggatott vonallal választjuk el egymástól, a téglalap bal felső sarkába a *par* szó kerül:



Az ATM-mel kapcsolatos példánál maradva, a gyakorlatot figyelembe véve egy bankban egyidejűleg több aktív ATM-kapcsolat lehet, és mindegyik kapcsolat tevékenységei párhuzamosan folynak. Ha a szekvenciadiagramban az ATM készülékeket összevontan kezeljük, és ezenkívül csak a bankot vesszük figyelembe, akkor a szekvenciadiagram:

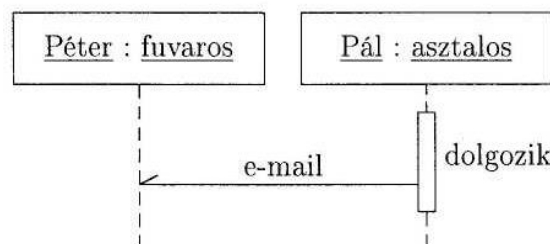


Példa:

Pál asztalos, Péter fuvaros. Pál hozzákezd az asztal elkészítéséhez. Amikor már biztosan tudja, hogy mikor fejezi be a munkát, e-mail üzenetben közli Péterrel a szállítás időpontját. Készítsük el a szöveg alapján a szekvenciadiagramot.

Megoldás:

Pál az asztalos osztály egy objektuma: Pál : asztalos. Hasonlóképpen: Péter : fuvaros. Pál esetében azt a szerepét, amikor az asztalt készíti, meg kell különböztetni attól, amikor nem készíti. Ezért Pál aktív szerepe legyen az, hogy asztalt készít. Pál életvonala során ebben az aktív szerepben – annak befejezése előtt – e-mail üzenetet küld Péternek. Az e-mail üzenet elküldése nem jár azzal a következménnyel, hogy Pálnak meg kellene szakítania a munkáját. Nem kerül blokkolt állapotba. Ez az üzenet tehát aszinkron üzenet. A szekvenciadiagram:



Példa:

Az óvodások tornagyakorlatát az egyik óvó néni, Jutka néni, sípjelekkel vezérli, miközben a másik óvó néni, Marika néni, zöld zászló magasba emelésével ad engedélyt az utasítás végrehajtására. Szabályok:

- Az óvodásoknak, ha ülnek, a sípjelre fel kell állniuk, de csak akkor, ha a zöld zászló is a magasban van.
- Az óvodásoknak, ha állnak, a sípjelre le kell ülniük, de csak akkor, ha a zöld zászló nincs a magasban.

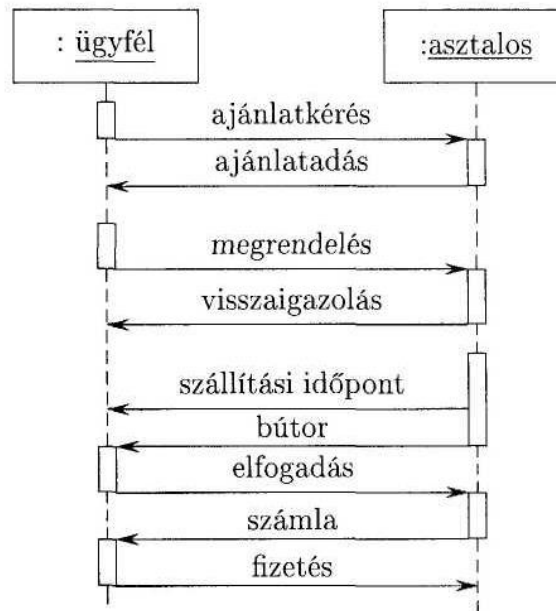
Készítsük el a játék menetét szemléltető szekvenciadiagramot.

Megoldás:

Elemzés:

- A játékot Jutka néni és Marika néni vezérli. Ők az aktorok.

A leírás alapján értelemszerűen a szekvenciadiagram:



Példa:

Készítsük el a kisvasút mozdonyának vezérlésére írandó program modelljét, ahol a programnak a következő követelményeket kell kielégítenie:

1. A mozdony motorját, sebességváltóját és lámpáját kell egy-egy mikroprogramnak vezérelnie.
2. A sebességváltó lehet üres állapotban, előremenetben vagy hátramenetben; de mind előremenetben, mind hátramenetben két sebességfokozatba kapcsolható.
3. A motor állandó sebességgel forog, ha áram alá helyezzük.
4. A lámpát akkor lehet csak bekapcsolni, ha a motor jár.
5. A rendszert a játékos kattintásokkal vezérli úgy, hogy külön gombot használ az előremenet, illetve a hátramenet vezérlésére.
6. Ha a játékos kattint egyet, akkor a kiválasztott gombnak megfelelő sebességfokozatba kapcsol, és elindítja a motort.
7. Azonos gombon történő újabb kattintásokkal a sebességfokozatokat váltogathatja.
8. Ellenkező irányhoz tartozó gombon történő kattintással a sebességváltó üresbe kerül, és a motor leáll. Ekkor a lámpa is elalszik, ha netán be volt kapcsolva.
9. A lámpát a két gomb egyidejű kattintásával lehet be- vagy kikapcsolni.

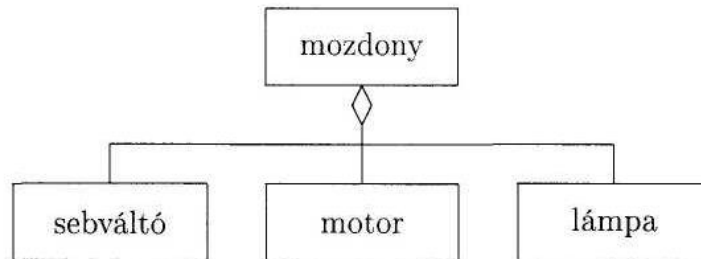
Készítsük el az osztálydiagramot, a szekvenciadiagramot és az állapotdiagramokat.

Megoldás:

Osztálydiagram:

Először a követelmények elemzése alapján határozzuk meg az objektumokat, amelyeket osztályokba foglalunk össze, és rajzoljuk meg az osztálydiagramot. Ehhez figyeljük meg a követelmények szövegét:

„A mozdony motorját, sebességváltóját és lámpáját kell egy-egy mikroprogramnak vezérelni.”
Itt tehát aggregációról van szó: a mozdonyt a sebváltó, a motor és a lámpa alkotja:



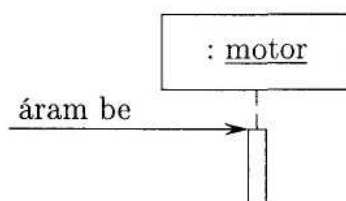
Szekvenciadiagram:

A statikus modell ismeretében készítsük el a követelmények elemzése alapján a szekvenciadiagramot a megoldás menetének szemléltetésére. Vegyük sorba a követelmények megfelelő pontjait, de előtte vezessük be az osztályszerkepek jelölésére a következőket:

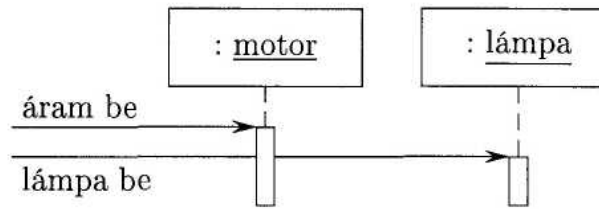
- : motor : az aktív rész jelöli, hogy a motor jár;
- : lámpa : az aktív rész jelöli, hogy a lámpa világít;
- előre₁ : a sebességváltó előremenet első fokozatban van;
- előre₂ : a sebességváltó előremenet második fokozatban van;
- üres : a sebességváltó üresben van;
- hátra₁ : a sebességváltó hátramenet első fokozatban van;
- hátra₂ : a sebességváltó hátramenet második fokozatban van.

A sebességváltó esetén a rövideg kedvéért csak a szerep nevét adjuk meg, az osztály megnevezését elhagyjuk.

„A motor állandó sebességgel forog, ha áram alá helyezzük”, azaz a motort az áram bekapcsolásának eseménye aktivizálja:



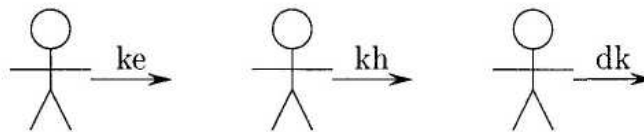
„A lámpát csak akkor lehet bekapcsolni, ha a motor jár.” A lámpát tehát csak olyan időpontban lehet aktivizálni, amikor a motor is aktív:



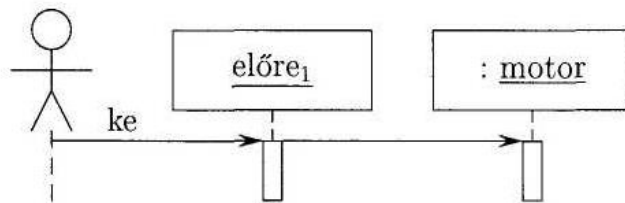
„A rendszert a játékos kattintásokkal vezérli úgy, hogy külön gombot használ az előremenet és a hátramenet vezérlésére.” Jelöljük ezeket az üzeneteket röviden:

- kattintás előre: ke,
- kattintás hátra: kh,
- a két gomb egyidejű kattintása : dk

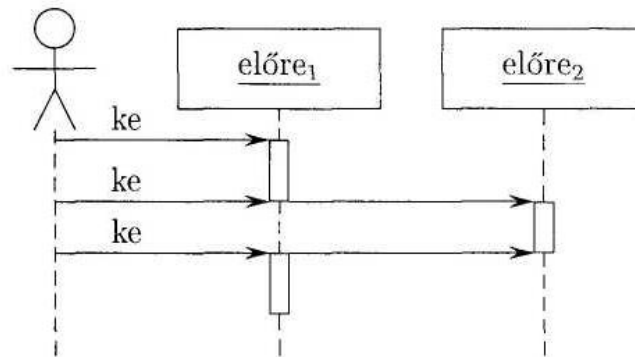
A játékos ezek szerint az aktor szerepét tölti be ezen üzenetek alatt:



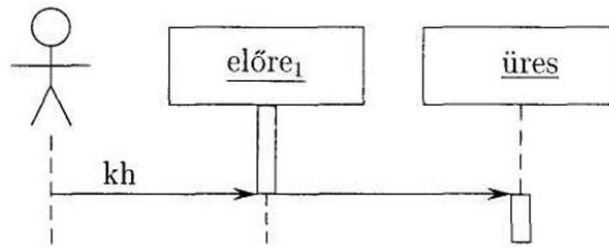
„Ha a játékos kattint egyet, akkor a kiválasztott gombnak megfelelő sebességfokozatba kapcsol, és elindítja a motort, ha az nem járt még.”:



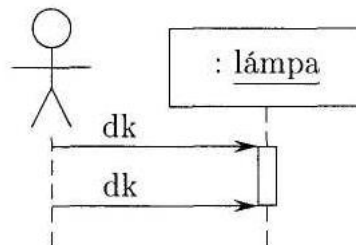
„Azonos gombon történő kattintásokkal a sebességfokozatokat változathatja a játékos.”:



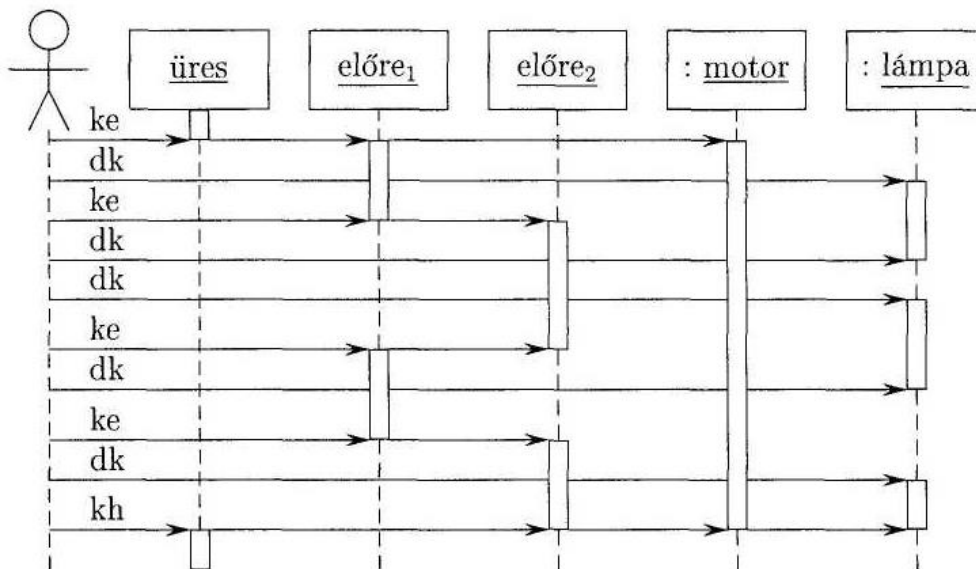
„Ellenkező irányhoz tartozó gombon történő kattintással a sebességváltó üresbe kerül, (és a motor leáll. Ekkor a lámpa is elalszik, ha netán be volt kapcsolva.)”:



„A lámpát a kettős kattintásokkal lehet be- vagy kikapcsolni.”:



Az eddigieket megértve és összevetve azt kapjuk, hogy a rendszer működését az alábbi ábrán látható diagram szemlélteti: (Az egyszerűség kedvéért eltekintettünk a hátrameneti állapotoktól, amelyek az előremeneti állapotokhoz hasonlóak.)

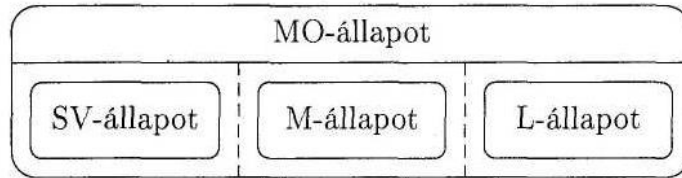


Állapotdiagram:

A mozdony állapotát (MO-állapot)

- a sebváltó állapota: SV-állapot,
- a motor állapota: M-állapot,
- és a lámpa állapota: L-állapot

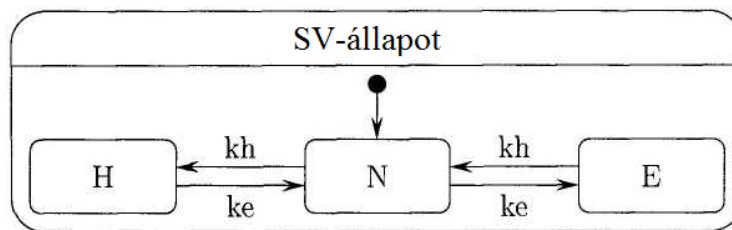
együttesen határozza meg. Ezért a mozdony állapota ezen állapotok aggregációja:



A sebességváltónak három állapota lehet:

- üres állapot : N,
- előremeneti állapot : E,
- hátrameneti állapot : H.

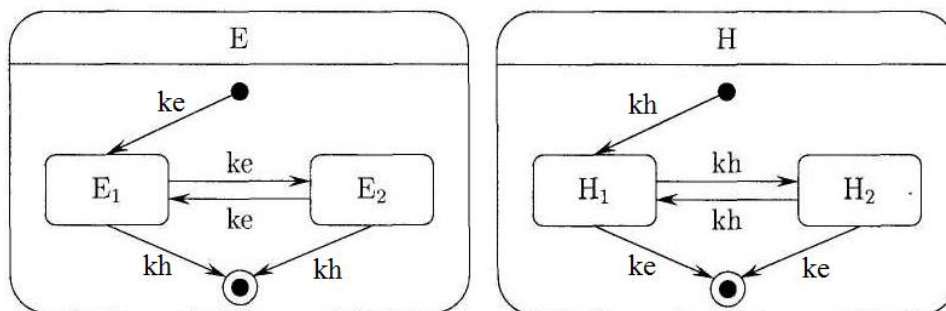
A sebességváltó állapota a fenti állapotok általánosítása, amelyek között az átmeneteket a megfelelő kattintások idézik elő:



Az előremeneti, valamint a hátrameneti általánosított állapotnak két-két konkrét állapota lehet:

- az előremeneti állapot (E) részállapotai:
 - előremenet 1. fokozatban : E_1 ,
 - előremenet 2. fokozatban : E_2 ;
- a hátrameneti állapot (H) részállapotai:
 - hátramenet 1. fokozatban : H_1 ,
 - hátramenet 2. fokozatban : H_2 .

Ennek megfelelően kapjuk az előremeneti és a hátrameneti állapotok állapotdiagramjait, figyelembe véve, hogy a részállapotok közötti átmeneteket a megfelelő kattintások váltják ki:



A motornak két állapota lehet:

- a motor áll : MÁ,

- a motor jár : MJ.

A motor állapotai közötti átmenetek:

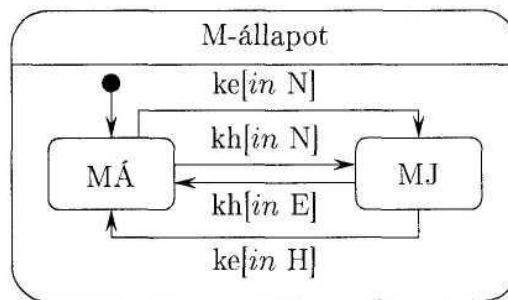
1. A motor elindításának feltétele, hogy egy kattintáskor sebességváltó üres állapotban van, azaz

- $kh[in\ N]$ vagy
- $ke[in\ N]$.

2. A motor megállításának feltétele, hogy egy kattintáskor sebességváltó az ellenkező irányban megfelelő állapotban van, azaz

- $kh[in\ E]$ vagy
- $ke[in\ H]$.

Ennek alapján az állapotdiagram:



A lámpának két állapota lehet:

- a lámpa be van kapcsolva : LBE,
- a lámpa ki van kapcsolva : LKI.

A lámpa állapotai közötti átmenetek:

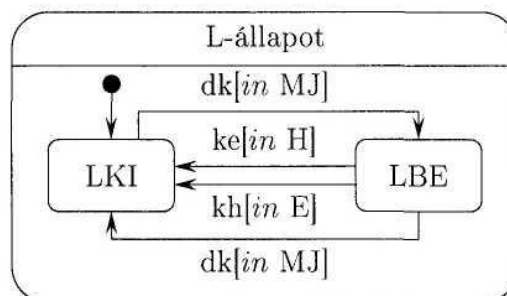
1. A lámpa bekapcsolásának feltétele:

- kettős kattintással a LKI állapotból lépünk ki, amikor a motor jár: $dk[in\ MJ]$.

2. A lámpa kikapcsolásának feltétele:

- kettős kattintással LBE állapotból lépünk ki, amikor a motor jár: $dk[in\ MJ]$;
- megszűnik az áram, mert a menetiránnyal ellentétes gombon kattintunk: $kh[in\ E]$ vagy $ke[in\ H]$.

Így az állapotdiagram:

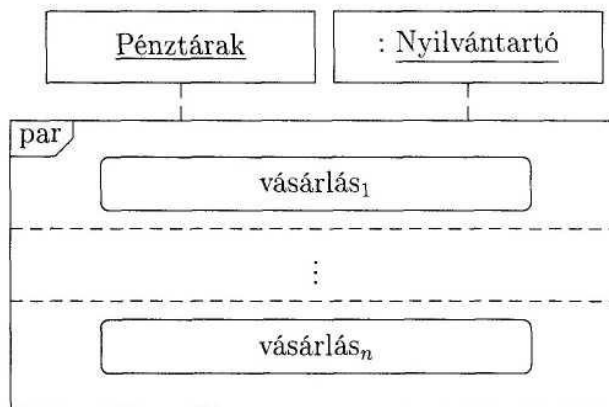


Példa:

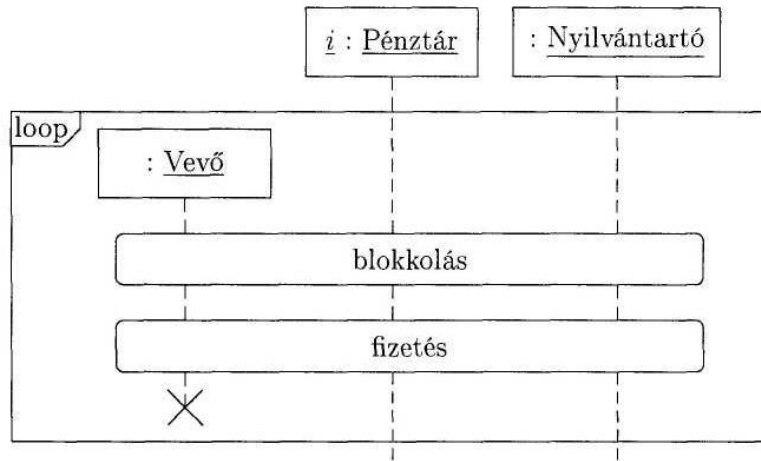
Készítsünk szekvenciadiagramot, amely a következő áruházi vásárlást modellezi. Az áruházban több pénztárnál fizethetnek a vevők, de egy vevő csak egy pénztárnál. A pénztárok párhuzamosan működnek. Egy vevő a kiválasztott pénztárnál felrakja az árukat a szalagra, ahonnan a pénztáros elveszi. A vonalkód alapján a központi nyilvántartásból megállapítja az áru nevét és árát, amit a blokkra nyomtat. Ha végzett a vevő összes árujával, akkor összegzi az árakat. Ezután a vevő fizet. A fizetés történhet készpénzzel vagy kártyával. Készpénzes fizetés esetén a vevő átadja a pénzt, és ha ez meghaladja a blokkon szereplő összeget, a pénztáros visszaad. Kártyás fizetés esetén a vevő átadja a kártyát, amit a pénztáros „lehúz”, azaz a banknak elküldi a kártyaszámot. Ezután megadja az összeget, amit a bank és a vevő is ellenőriz. Ha rendben van, akkor a vevő megadja a PIN-kódot, amit a bank ellenőriz, és az ellenőrzés eredményét továbbítja a pénztárosnak. Ha a kód rendben van, a pénztáros elismervényt nyomtat, amit átad vevőnek. Ezt a vevő aláírva visszaadja. Ezután a pénztáros átadja az elismervény másolatát. Mindkét fizetési mód esetén a pénztáros a blokkot is átadja. Ezután a pénztáros a következő vevővel foglalkozik.

Megoldás:

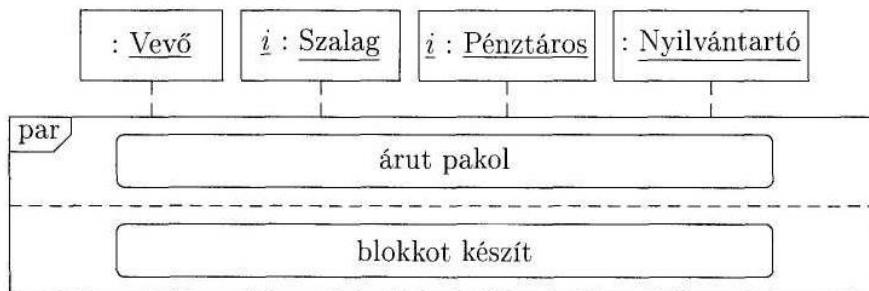
Ha a pénztárat összevonva kezeljük, akkor két szerepet különíthetünk el: az (árúkészlet és ár) nyilvántartót, illetve a pénztárat. Ha a működő pénztárok száma n , akkor n vásárlási tevékenység zajlik párhuzamosan, azaz a szekvenciadiagram legfelső szintje:



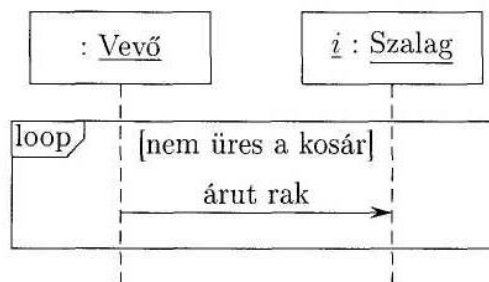
Ezt a diagramot kell finomítani, amelynek során példány dekompozícióval meg kell adnunk a pénztárat részletesen, másrészt az egyes vásárlási tevékenységeket kell pontosítani. Az i . vásárlás pontosítása során a pénztárat fel kell bontani az i . pénztárra és az ott található vevőkre. A tevékenység egy ciklus, amelyben a pénztár sorban szolgálja ki a vevőket. Egy kiszolgálás két nagy részre osztható: blokkolásra és fizetésre. Egy adott cikluson belül mindig egy vevővel foglalkozunk, aki tulajdonképpen ekkor „jön létre” és „szűnik meg”:



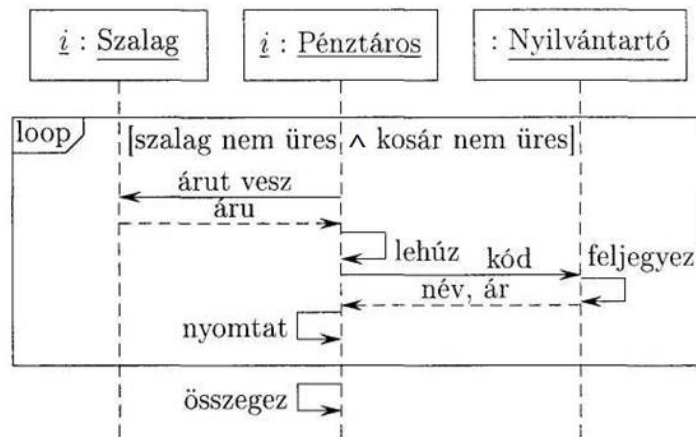
A blokkolás során a pénztárt tovább bontjuk egy pénztárosra és egy szalagra. A tevékenység során a vevő és a pénztáros párhuzamosan dolgozhat. A vevő a szalagra pakol, a pénztáros a szalagról veszi az árukat és blokkol:



Az áruk pakolása egy ciklus, amelyben a vevő árut rak a szalagra, amíg ki nem ürül a kosara:

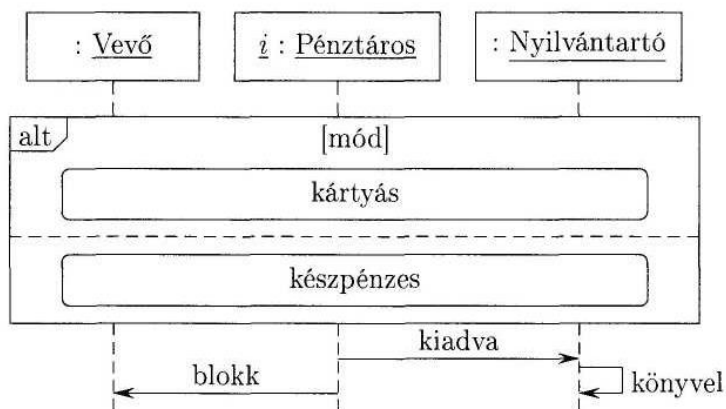


A pénztáros, amikor blokkot készít, egy ciklusban árukat vesz a szalagról. Az áru vonalkódját leolvassa, és a kódot a nyilvántartóhoz továbbítja. Innen megkapja a nevet és az árat, amit kinyomtat a blokkra. Közben a nyilvántartó feljegyzi az árucikk kódját, hogy a vásárlás végén csökkenteni tudja a készletet. Az iteráció addig tart, amíg a pénztáros a vevő összes áruját fel nem dolgozza, azaz amíg a kosár és a szalag nem üres. Ezután egy összegzést készít:

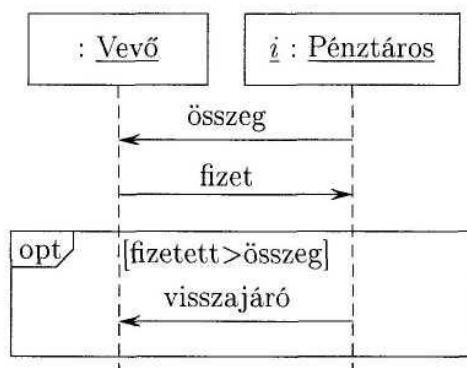


Az egyszerűség érdekében nem vettük figyelembe az esetleges hibákat, amelyek például egy áru lehúzásakor keletkezhetnek. A továbbiakban sem kezeljük a hibás eseteket.

A fizetés lehet kártyás vagy készpénzes. Mindkét esetben az összeg átvétele után a pénztáros üzenet küld a nyilvántartónak, hogy az áruk tényleg ki lettek adva, majd a vevőnek átadja a blokkot. A nyilvántartó a feljegyzések alapján módosítja a készletet:

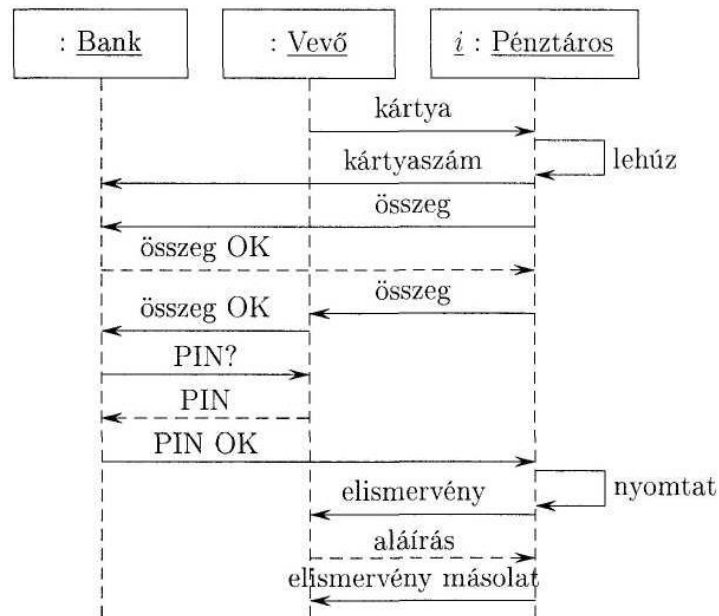


Készpénzes fizetés esetén a pénztáros megadja az összeget, ezután a vevő fizet. Ha a kifizetett összeg nagyobb, mint a blokkon szereplő összeg, akkor a visszajárót átadja a pénztáros:



Kártyás fizetés során a vevő átadja a kártyát. Ezt a pénztáros lehúzza, és a kártyaszámot elküldi a banknak. Ezután az összeget is, amelyről a bank eldönti, hogy kifizethető-e. Majd a pénztáros megadja a vevőnek az összeget, amit az ellenőriz, és egy gomb megnyomásával nyugtáz a bank felé. Ezután a vevő megadja a banknak a PIN-kódot, amit az ellenőriz, és erről

értésíti a pénztárost. A pénztáros kinyomtatja az elismervényt és átadja azt a vevőnek. A vevő ezt aláírva visszaadja. Végül a pénztáros átadja az elismervény másolatát:



Az olvasó feladata végiggondolni, hogy az esetleges hibák, rendellenességek kezelésével miként lehet a diagramokat kiegészíteni. A következőkben megadunk néhány lehetséges hibaesetet, amelyeket egy valós rendszerben figyelembe kell venni:

- Az áru lehúzásakor nem sikerül a vonalkódot módosítani.
- Nincs elég készpénz a vevőnél, és árukat kell „visszavenni”, amíg az összeg már megfelelő.
- Kártyás fizetés esetén nem sikerül a kártyát leolvasni, és néhányszor újra kell próbálkozni.
- Az összeg túllépi a kártyával fizethető maximális összeget, és vissza kell venni árukat, vagy nem lehet kártyával fizetni.
- Nem jó a megadott PIN-kód, és újra kell próbálkozni.
- Ha nem lehet kártyával fizetni, akkor meg kell próbálni a készpénzes fizetést. Ha arra nincs lehetőség, az egész vásárlást törölni kell.