

Példa: pénzkiadó automata

Ficsor Lajos
Miskolci Egyetem
Általános Informatikai Tanszék
Utolsó módosítás: 2005. 11. 30.

A feladat

- Készítsünk Java alkalmazást, amely egy ATM-et modellez. Az ügyfélnek van egy bankszámlája és egy bankkártyája. A kártya szerepel a számla és a PIN kód. Az ügyfél behelyezi a kártyáját az automatába, majd beadja a PIN kódot. Az ATM ellenőrzi a PIN helyességét, és helytelen kód esetén kidobja a kártyát. Egyébként bekéri a felvenni kívánt összeget. Ha van elég pénz az automatában és az ügyfél számláján is, leemeli a számláról az összeget, egyébként kidobja a kártyát.

A feladat elemzése

- A fenti feladat egy egyszerűsített leírás.
- Ebből a leírásból kell felépítenünk az analízis modellt.
- Lépések:
 - Osztályok azonosítása
 - Osztályok közötti kapcsolatok azonosítása
 - Osztályok attribútumainak azonosítása (adattagok)
 - Osztályok operációinak azonosítása (metódusok)

Osztályok azonosítása

- Legtöbbször főnevek utalnak rá.
- Csak azt érdemes osztályként modellezni, aminek van állapota és viselkedése, tehát nem minden főnév jelez osztályt.
- Lehetnek olyan főnevek is, amelyek nem a rendszerhez tartozó dolgokat jelölnek.
- Különböző szavak jelölhetik ugyanazt a fogalmat.
- A következő oldalon kiemeltük a legfontosabb főneveket.

Osztályok azonosítása (folyt.)

- Készítsünk Java **alkalmazást**, amely egy **ATM-et** modellez. Az **ügyfélnek** van egy **bankszámlája** és egy **bankkártyája**. A kártyán szerepel a számla és a **PIN** kód. Az ügyfél behelyezi a kártyáját az **automatába**, majd beadja a PIN kódot. Az ATM ellenőrzi a PIN helyességét, és helytelen kód esetén kidobja a kártyát. Egyébként bekéri a felvenni kívánt **összeget**. Ha van elég **pénz** az automatában és az ügyfél számláján is, leemeli a számláról az összeget, egyébként kidobja a kártyát.

Osztályok azonosítása (folyt.)

- Elemezzük az eredményt:

alkalmazás	Maga a rendszer. (Egyelőre nem osztály)
ATM	Osztály.
ügyfél	Külső elem, nem a rendszer része.
bankszámla	Osztály.
bankkártya	Osztály.
automata	Az ATM szinonímája, nem új osztály.
összeg	Csak adat.
pénz	Csak adat.

Első osztálydiagram

Automata

Bankkartya

Bankszaml

Osztályok közötti kapcsolatok

- Mely osztályok állnak egymással kapcsolatban (például használják egymás adatait vagy módszereit)?
- Mi a kapcsolat iránya?
- A példánkban:
 - A Bankkartya osztály kapcsolatban van a Bankszaml osztállyal, mert ismerni kell az adatait. Fordítva nem igaz.
 - Az Automata osztálynak ismernie kell a Bankkartya osztályt, fordítva nem igaz.

Osztálydiagram kapcsolatokkal

Automata

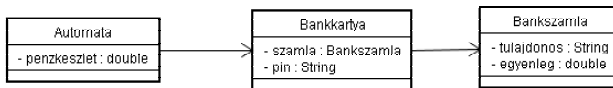
Bankkartya

Bankszaml

Osztályok attribútumai

- Minden osztályra meg kell vizsgálni. A szövegből vagy egyéb információkból következhetnek.
- Például a Bankkartya osztályra:
 - "A kártyán szerepel a számla és a **PIN** kód."
- A Bankszamla osztályra:
 - "Az **ügyfélnek** van egy **bankszámlája ...**"
 - "Ha van elég **pénz ...** az ügyfél számláján ..."
- Az Automata osztályra:
 - "Ha van elég **pénz** az automatában..."

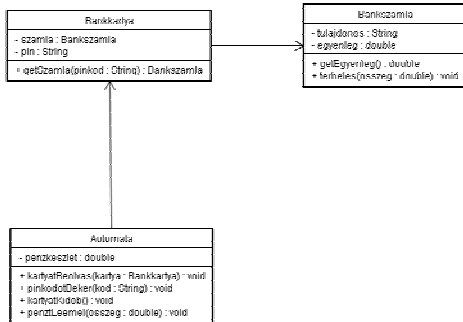
Osztálydiagram attribútumokkal



Osztályok operációi

- Minden osztályra meg kell vizsgálni, hogy milyen viselkedés mintákat várunk el tőlük.
- Ezek lesznek az osztály nyilvános metódusai.
- Például Bankszamla meg kell hogy tudja mondani és csökkentenie kell tudnia az egyenlegét.
- Az Automata legfontosabb műveletei:
 - kártyát beolvas
 - PIN kódot beolvas
 - összeget bekér
 - az összeget leemeli a számláról.

Osztálydiagram operációkkal



A modell ellenőrzése

- Minden, a működéshez szükséges kapcsolat megvan?
- Minden operáció és attribútum megvan?



A modell kiegészítése

- A modellt kiegészítjük az implementáláshoz szükséges részletekkel. Például:
 - konstruktorok
 - private metódusok.



Mitől kezd el működni?

- Az osztályokból objektumokat kell példányosítani.
- Ezek után a megfelelő metódushívások hajtják végre a megfelelő lépéseket.
- Az objektumokat csoportosítani kell az alábbiak szerint:
 - aktív: példányosítani tud és üzenetet tud küldeni más objektumoknak
 - passzív: ha meghívják egy metódusát, azt végrehajtja.

Mitől kezd el működni? (folyt.)

- Az egyszerű eset, ha pontosan egy aktív objektum van az alkalmazásban.
 - Ilyenkor egy main metódus példányosítja ezt az objektumot, és meghívja egy metódusát. Innen már minden megy "magától".
- Probléma:
 - több aktív objektum: párhuzamos működést kell implementálni (szálkezelés)
 - egy aktív objektum sincs: ilyenkor létre kell egyet hozni. ("Alkalmazás" objektum.)

Mitől kezd el működni? (folyt.)

- A példában mindhárom objektum passzív, tehát kell egy "Alkalmazás" objektum.

Az implementáció

- A modellt le kell fordítani Java-ra.
- Ebben a kódgenerálás segíthet.
- A példa osztálydiagramja a Jude fejlesztő eszközzel készült, amely képes kód generálásra.
- Ezt persze a metódusok implementálásával ki kell egészíteni.
- Például a Bankkartya osztály generált kódja

Bankkartya osztály generált kódja

```
package atml;  
  
public class Bankkartya {  
    private Bankszamlam szamlam;  
    private String pin;  
    public Bankszamlam bankszamlam;  
    public Bankszamlam getSzamlam(String  
        pinkod) {  
        return null;  
    }  
}
```
