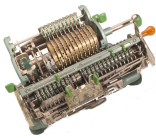


Szoftver-technológia II.

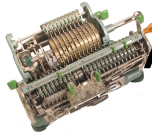
Tervezési minták



Szoftver-technológia II.

Irodalom

- Steven R. Schach: Object Oriented & Classical Software Engineering, McGRAW-HILL, 6th edition, 2005, chapter 8.
- E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design patterns: Elements of reusable object-oriented software, Addison-Wesley, 1995

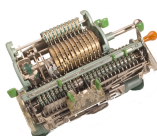


Szoftver-technológia II.

A szoftver újra felhasználás két alaptípusa

- Opportunista (esetleges) újra felhasználás
 - először a termék kerül kifejlesztésre, majd a komponensek bekerülnek az újra felhasználási adatbázisba
- Szisztematikus (szándékos) újra felh.
 - újra felhasználható komponensek létrehozása, majd a szoftver kifejlesztése

3

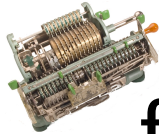


Szoftver-technológia II.

Miért kell?

- Termék gyorsabb piacra kerülése
- A létrehozott kód mindössze 15%-a új, eredeti

4

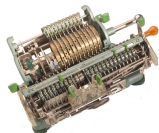


Szoftver-technológia I.

Az újra felhasználhatóság ára

- újra felhasználható elemek létrehozási költsége
- az elem újra felhasználásának költsége
- az újra felhasználási folyamat kialakításának és működtetésének költsége

5

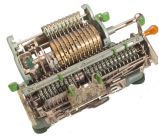


Szoftver-technológia II.

Tanulság

- Számtalan tanulmány szerint a szoftverfejlesztő szervezetekben alacsony az újra felhasználás foka
- nehéz jól, hatékonyan, széleskörűen újra felhasználható szoftver komponenseket előállítani

6

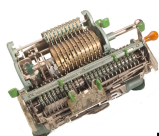


Szoftver-technológia II.

Struktúrált kód újra felhasználása

- Megfelelő modul kohézió esetén, a kód csak ugyanolyan adatokon, adatszerkezeteken használható fel újra

7

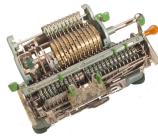


Szoftver-technológia II.

OOP újra felhasználhatóság

- Az objektumok, mivel biztosítják a logikai kohéziót, újra felhasználhatók

8

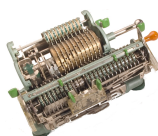


Szoftver-technológia II.

Újra felhasználási lehetőségek

- tervezési elemek újra felhasználása
- implementálási elemek újra felhasználása

9

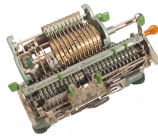


Szoftver-technológia II.

Tervezési újra felhasználás

- Opportunista újra felhasználás adott alkalmazási területhez kötődő szoftverek fejlesztése esetén

10

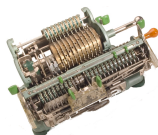


Szoftver-technológia II.

Újra felhasználási lehetőségek

- közvetlen kód, elemek
 - library-k, toolkitok, (függvény könyvtárak, osztály könyvtárak)
- alkalmazás keretrendszerek (application framework)
- tervezési minták (design pattern)

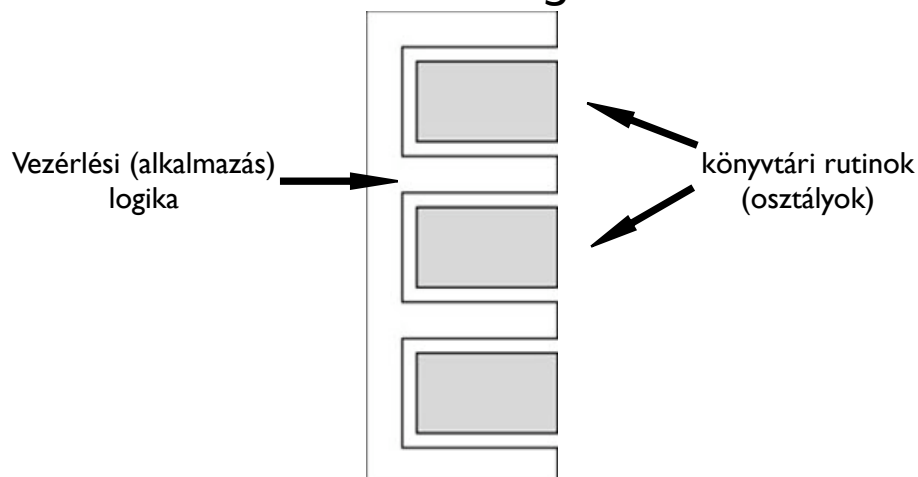
11



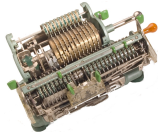
Szoftver-technológia II.

Library, toolkit

- Újra felhasználható rutinok
- Változó vezérlési logika



12

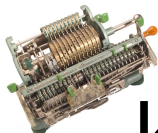


Szoftver-technológia II.

Alkalmazás keretrendszerek

- Komponensek integrált halmaza, melyek együttműködve egy újra felhasználható architektúrát eredményeznek, lehetővé teszik azonos jellegű alkalmazások létrehozását

15

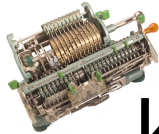


Szoftver-technológia II.

Alkalmazás keretrendszerek (folyt.)

- Gyorsabb fejlesztés mint a toolkitek esetén
 - a terv (architektúra) nagyobb része kerül újra felhasználásra
 - általában a működési logika létrehozása nehezebb, mint egy-egy műveleté

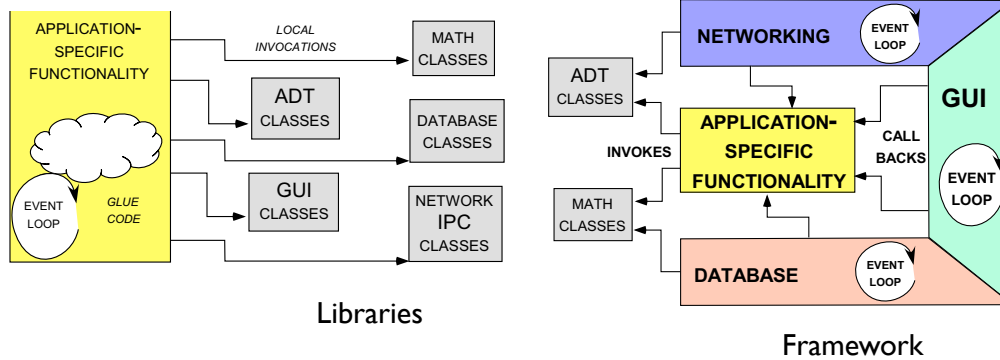
16



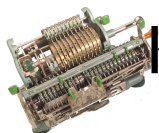
Szoftver-technológia II.

Alkalmazás keretrendszerek (folyt.)

- Libraries vs. Frameworks



17

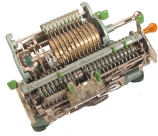


Szoftver-technológia II.

Komponensek integrálása a keretrendszerekbe

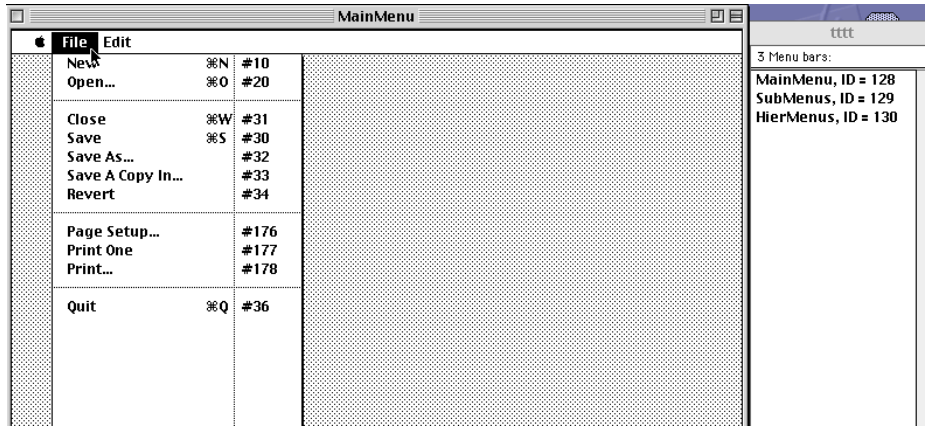
- Komponensek lazán csatoltak
 - callback interfész
- Függetlenül fejlesztett komponensek csatlakozathatók
- Általános (generikus) keretrendszer objektumok és alkalmazás specifikus objektumok

18



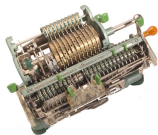
Szoftver-technológia II.

Keretrendszer példa



MacApp Framework

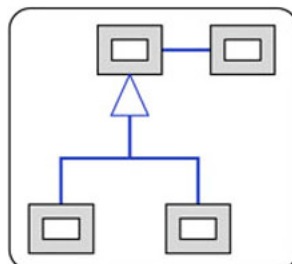
19



Szoftver-technológia II.

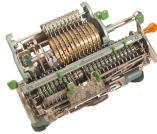
Tervezési minták

- A tervezési minták egy adott területen korábban sikeres megoldások szerkezetét, működését (invariáns értékeit) írják le



Mintában leírt szerkezet,
újra felhasználáskor a
konkrét osztályokat meg
kell adni

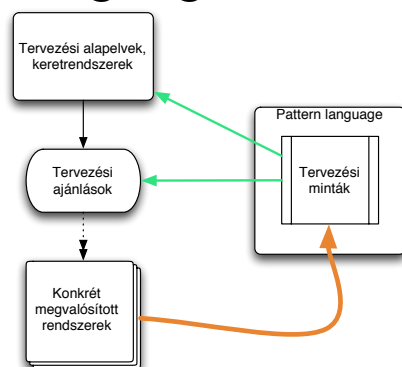
20



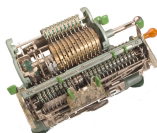
Szoftver-technológia II.

Tervezési minták (folyt.)

- Probléma és a megoldás lényegének leírása
 - tervezési szótár (pattern language)
 - azonosított minták
 - struktúrált
 - invariáns értékek
 - formális dokumentáció



21

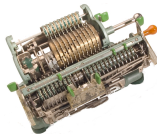


Szoftver-technológia II.

Minta leírása

- Név, cél
- Probléma és kontextus
- Szerkezet absztrakt leírása
- A használat pozitív és negatív következményei
- Implementációs irányelvek, kód példa
- Ismert felhasználások, kapcsolódó más minták

22

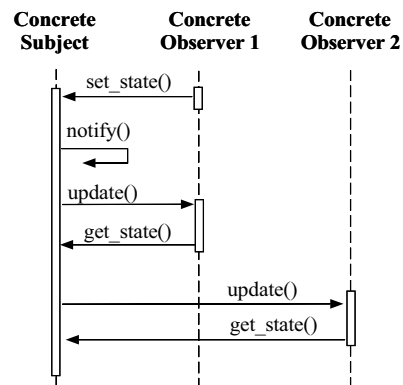
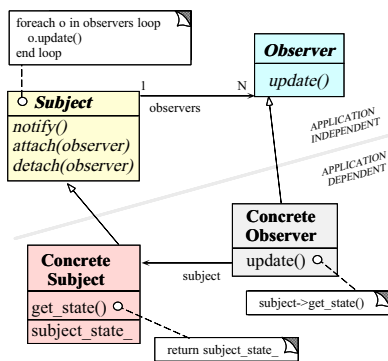


Szoftver-technológia II.

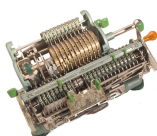
Tervezési minta (pl.)

- Observer pattern

Cél:
egy objektum állapotának
megváltozása több másik
objektumot érdekel



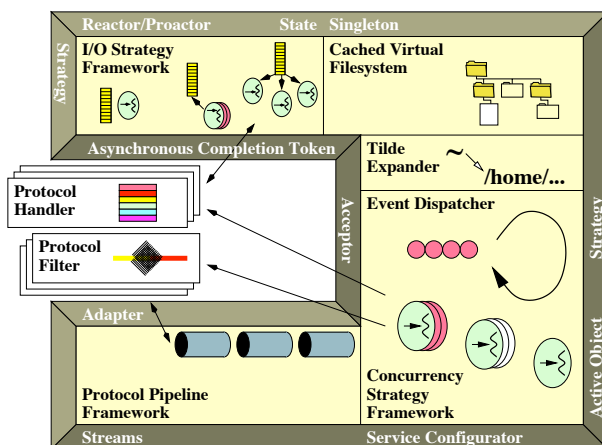
23



Szoftver-technológia II.

Tervezési minták (folyt.)

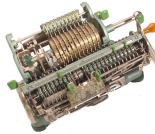
- Tervezési minta vs. alk. keretrendszer



Együtt léteznek
Minták - absztrakt leírások
Keretrendszerek - konkrét
nyelvi megvalósítások

A keretrendszerekben sok
minta jelenik meg

24

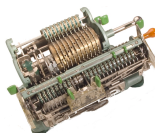


Szoftver-technológia II.

Minták használata

- Kisebb változtatásokkal előforduló, ismétlődő problémák megoldása
- Sok, bonyolultan kapcsolódó lépésből álló megoldást kívánó problémák
- Inkább belátást, megértést követelő helyzetek (probléma struktúrálás)

25

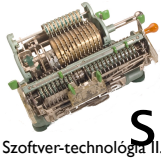


Szoftver-technológia II.

Mitől jó egy minta?

- Megold valamilyen problémát
 - hasznos
- Valamilyen meghatározott kontextusban érvényes
 - világosan kiderül hol használható
- Ismétlődik
 - újra felhasználható
- Tanít valamit
 - rávezet a probléma megoldására
- “Jó” neve van

26

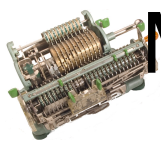


Szoftver-technológia II.

A minták használatához szükséges megközelítésmód

- Interfészek és implementáció elkülönítése
- Közös (állandó, stabil) és változó részek elkülönítése
- Helyettesíthetőség, változtathatóság megvalósítása

27

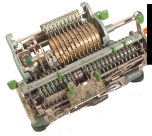


Szoftver-technológia II.

Minták használatának előnyei

- Szoftver architektúrák kiterjedt újra felhasználása
 - rendszerek hatékony dokumentálása
- Szakértői tudás rögzítése, elérhetővé tétele
- Fejlesztői kommunikáció támogatása
- Áttérés támogatása OOP-ra

28

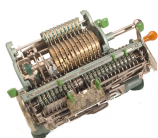


Szoftver-technológia II.

Minták használatának hátrányai

- Tervezési minták nem vezetnek közvetlen kód újra felhasználáshoz
- Minták félrevezetően egyszerűnek tűnhetnek
- Túl sok alkalmazandó minta megjelenése
- Minták automatizált teszteléssel nem validálhatók
- Minták alkalmazása emberi munka igényes

29

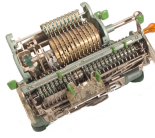


Szoftver-technológia II.

Hatékony minta alkalmazás

- Ne tervezzünk újra mindent mintának
- Támogassuk a minták kifejlesztését
- A minták kifejlesztésébe vonjuk be a domén szakértőket és alkalmazás fejlesztőket
- Világosan dokumentáljuk a minták alkalmazásának korlátait

30



Szoftver-technológia II.

Újra felhasználással adódó szoftver architektúra

- Könyvtárak
- Alkalmazás keretrendszer
- Tervezési minták

