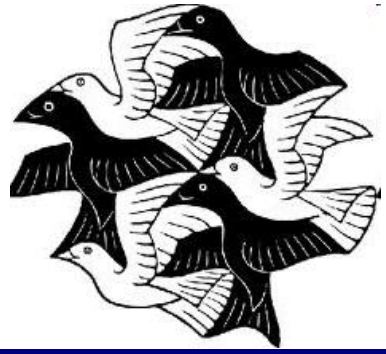


2. Въведение в шаблоните за проектиране

Софтуерни шаблони за проектиране

Боян Бончев,
ФМИ – Софийски университет

Анотация

- Произход и история на шаблоните
- Дефиниции
- Свойства
- Типове шаблони
- Описание на шаблоните за проектиране
- Език, система и каталог от шаблони
- Реорганизация на ОО приложения чрез шаблони за проектиране
- GoF шаблони

Използвана литература

- Gamma, Helm, Johnson, Vlissides **("Gang of Four“ - GoF**) *Design Patterns: Elements of Reusable Object-Oriented Software*, 1995
- *THE DESIGN PATTERNS JAVA COMPANION*, by James W. Cooper, Addison-Wesley, October 2, 1998
- *Design Patterns Explained*, by Allan Shalloway and James Trott, Prentice Hall, 2001
- *Thinking in Patterns*, by Bruce Eckel, Revision 0.9, 5-20-2003
- James O. Coplien (June 1996). *Software Patterns*. ISBN 978-1884842504

Произход и история на шаблоните 1/3

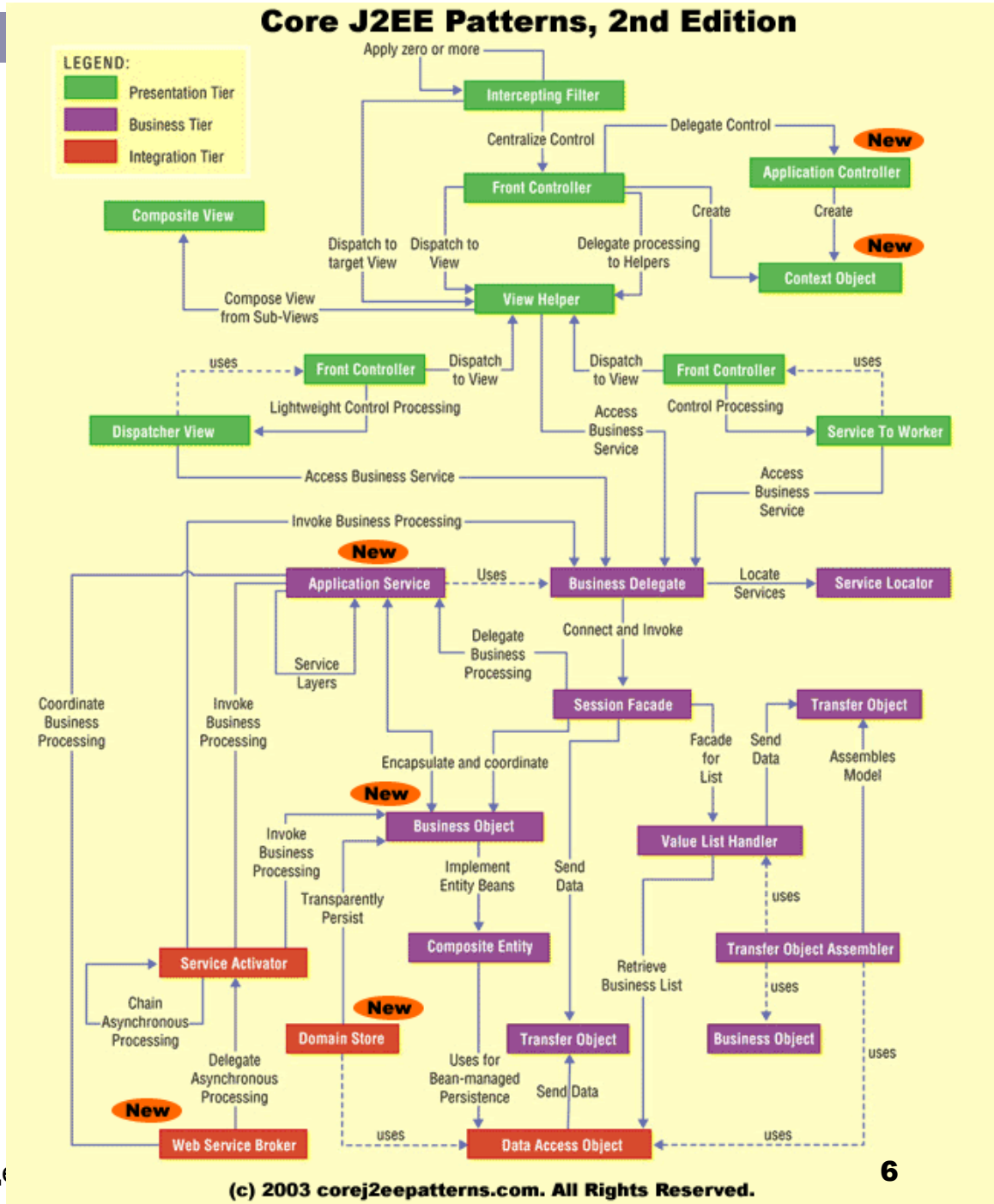
- Статии на архитекта Christopher Alexander **(първи въвел термина шаблон (pattern) през 1977-1979) – в** градското планиране и жилищната и обществена архитектура
- Kent Beck и Ward Cunningham, Textronix, в статия за OOPSLA'87 (**използват идеите за шаблони на Alexander за дизайн на Smalltalk GUI**)
- Erich Gamma, дисертация, 1988-1991
- James Coplien, в книгата *Advanced C++*, 1989-1991

Произход и история на шаблоните 2/3

- Erich Gamma, Richard Helm, Ralph Johnson, и John Vlissides **("Gang of Four“ - GoF**): *Design Patterns: Elements of Reusable Object-Oriented Software*, 1991-1994
- Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal (**Gang of Five**): Pattern-Oriented Software Architecture: A System of Patterns (**известна като POSA**)
- Езици от шаблони за проектиране на софтуер

Произход и история на шаблоните 3/3

- J2EE *шаблони за проектиране*
- .NET *шаблони за проектиране*
- SOA *шаблони - http://www.soapatterns.org*



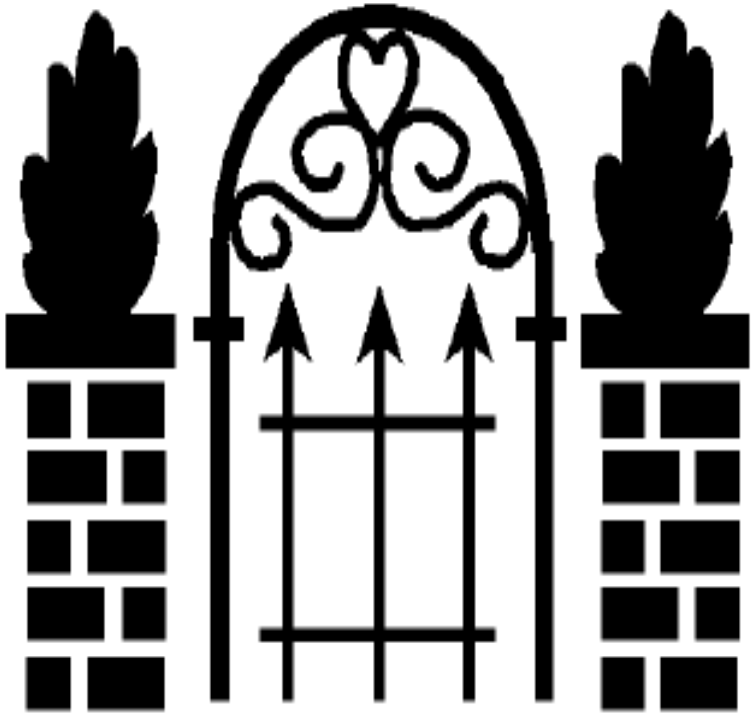
Дефиниции 1/2

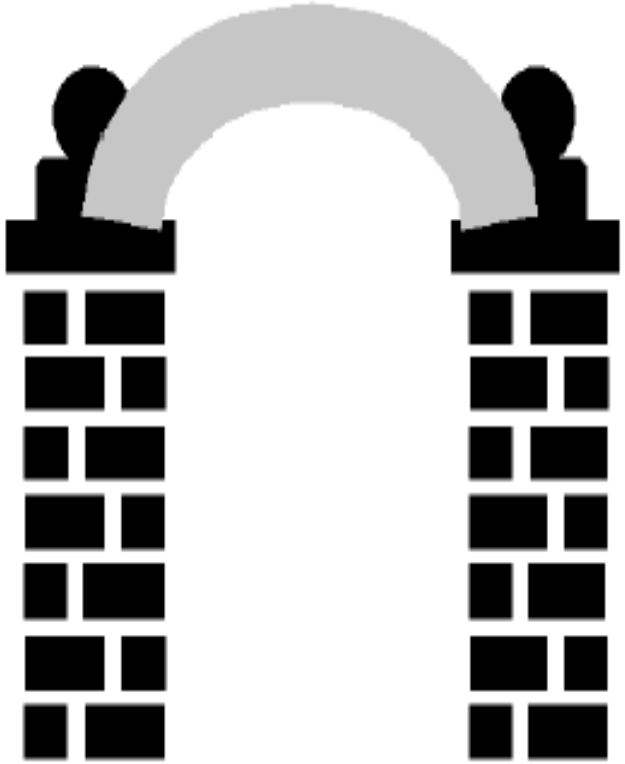
- … шаблонът за проектиране е *общо и многократно използваемо решение* на често срещани проблеми в дизайна
- Един шаблон за проектиране не е завършен дизайн, който може да се трансформира директно в реализация…
- *… описва проблем, който се появява отново и отново в нашето обкръжение, като задава основата на решение на този проблем по такъв начин, че можете да използвате решението един милион пъти, без да правите това по един и същ начин два пъти* **[Alexander]**
- … абстракция от конкретната форма, която поддържа в повтарящи се специфични и непроизволни контексти **[Riehle&Zullighoven**, Understanding and Using Patterns in Software Development**]**

Дефиниции 2/2

- …едновременно нещо и инструкции за неговото създаване **[Coplien,** Software Design Patterns: Common questions and Answers**]**
- …шаблонът може да се разглежда като особена форма на проза за запис на информация за дизайна, така че дизайните, които са работили добре в миналото, може да се прилагат отново в подобни ситуации в бъдеще **[Beck&Coplien,** Industrial experience with Design Patterns**]**
- ... литературен формат за улавяне на мъдростта и опита на експерти дизайнери и комуникация към начинаещите

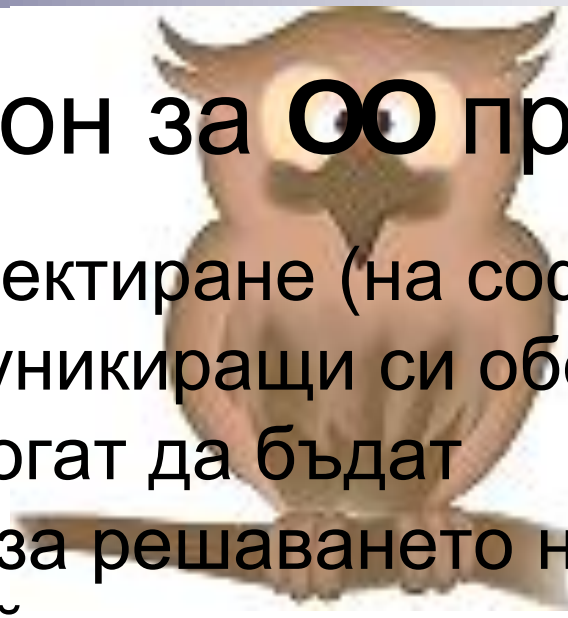
Архитектурни шаблони





Структурите могат да изглеждат различно, но да решават един и същ проблем

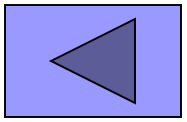
Какво е шаблон за ОО проектиране



- Шаблонът за проектиране (на софтуер) е описание на комуникиращи си обекти и класове, които могат да бъдат (пре)използвани за решаването на общ проблем на дизайна в конкретен контекст
- Шаблонът се състои от 4 основни елемента:
  - име
  - проблем
  - решение
  - последствия

Име на шаблон за проектиране

- Накратко описва проблем на дизайна и решението му
- Използва се между колеги, когато се говори за шаблона за проектиране
- Използвано е в документацията
- Подобрява използвания за проектирането речник
- Трябва да е кохерентно с останалите и навеждащо на подходящи асоциации



Проблем

- Описва кога се прилага шаблонът
- Обяснява проблема и контекста му
- Понякога задава списък от условия, които трябва да се спазят, за да има смисъл да се приложи шаблонът
- Трябва да се повтаря многократно в нашето работно обкръжение

Решение

- Описва елементите, които изграждат дизайна, и техните:
  - Взаимоотношения,
  - Отговорности
  - Кооперирane (collaboration)
- Не описва конкретен проект или имплементация
- Трябва да бъде добре доказано в реални проекти

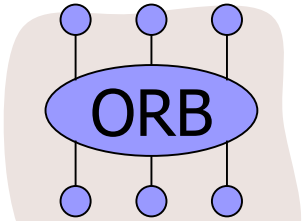
Последици

- Резултати и компромиси от прилагането на шаблона
- Полезни са за:
  - *Описание на решенията на дизайна*
  - *Оценяване на алтернативите за проектиране*
- Ползи от прилагането на шаблона
- Влияние върху системните *гъвкавост, разширяемост и преносимост*

Седемте нива на софтуерната архитектура*

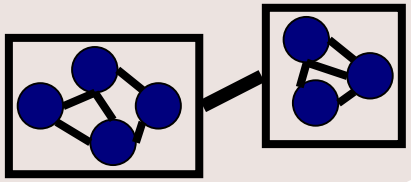
Глобална архитектура

Корпоративна архитектура

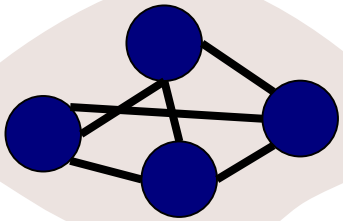


Системна архитектура

Подсистеми



Приложна архитектура

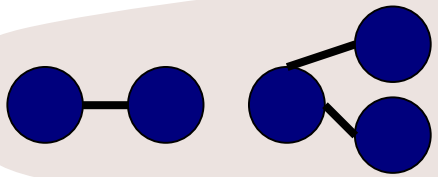


Приложни рамки

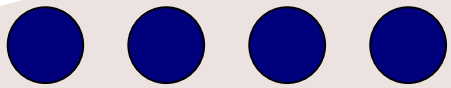
Макро-архитектура

шаблони за проектиране

Микро-архитектура



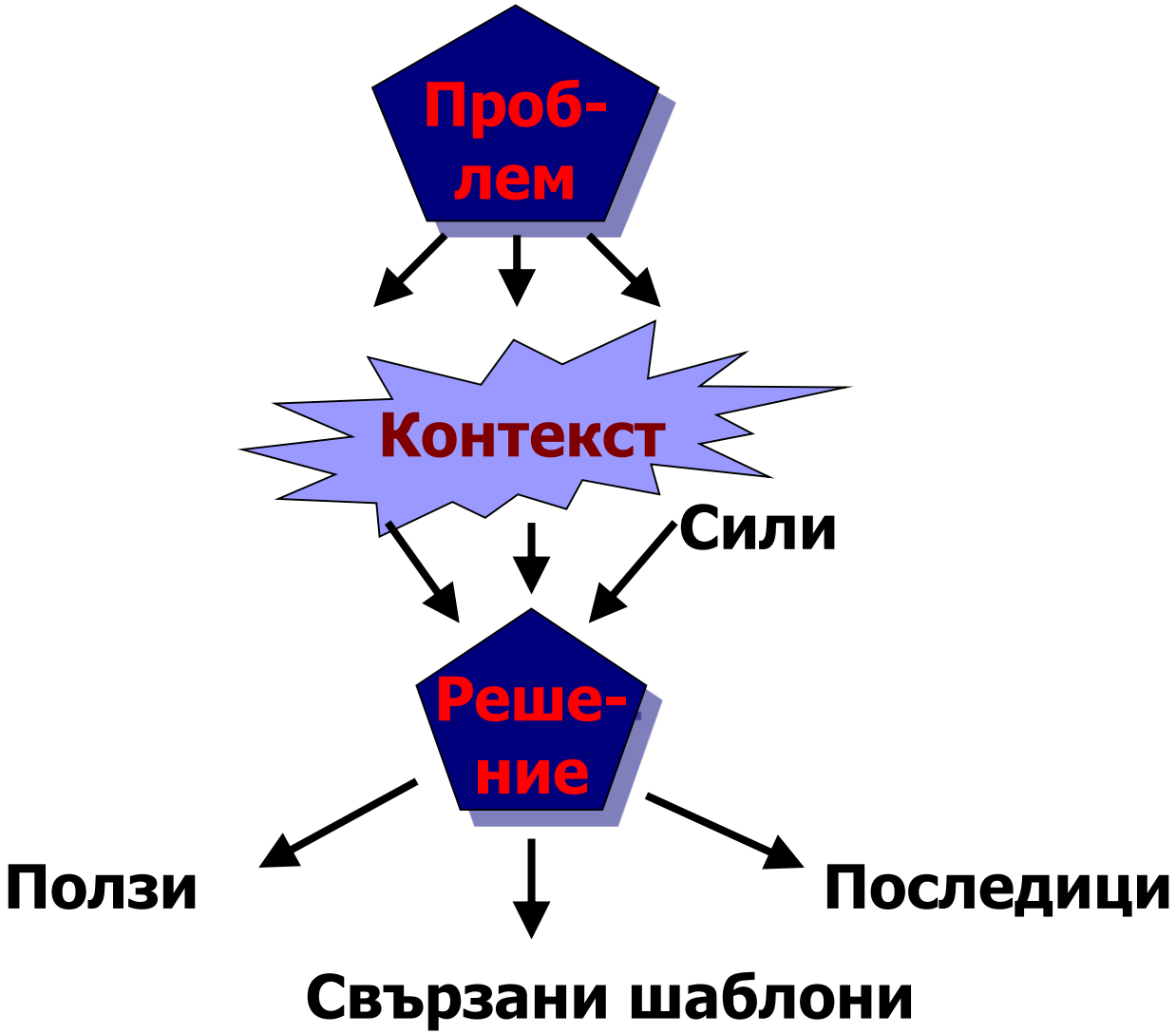
Обекти



ОО програмиране

* Mowbray and Malveau, 1997

Как възникват шаблоните?



Свойства – шаблоните:

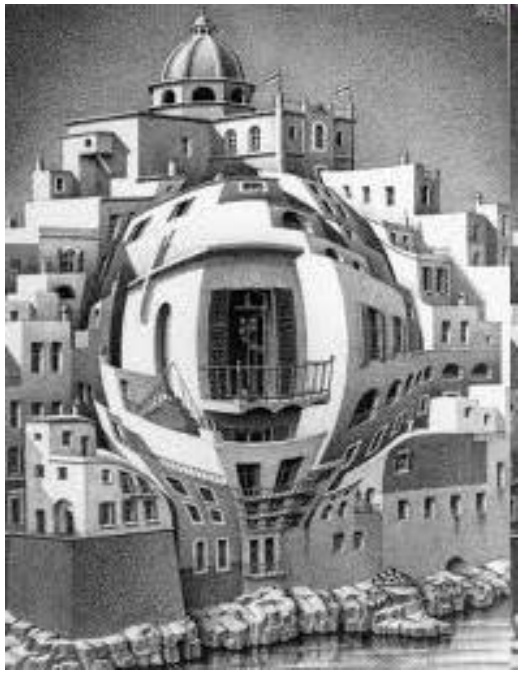
- Осигуряват общ речник за дизайнерите за споделяне, документиране и проучване на алтернативи за дизайна (Gamma'93)
- Описват софтуерните абстракции, осигуряват широка приложимост, и могат да предоставят съвети за потенциални проблеми при прилагането на сложни принципи
- Помагат за документиране на софтуерната архитектура
- Заснемат основните части на системния дизайн в компактна форма
- Показват повече от едно решение
- Описват софтуерни абстракции

Свойства – шаблоните **не**:

- предоставят имплементация
- осигуряват точното решение
- решават всички проектни проблеми
- се прилагат само за обектно-ориентиран дизайн

Шаблони навсякъде

- шаблоните обхващат всички аспекти на софтуерното инженерство, включително:
  - организационно развитие
  - процеси на разработка на софтуер
  - планиране на проекта
  - инженерингови изисквания
  - конфигурацията на софтуера
  - управление на проекта
  - .....



Типове софтуерни шаблони

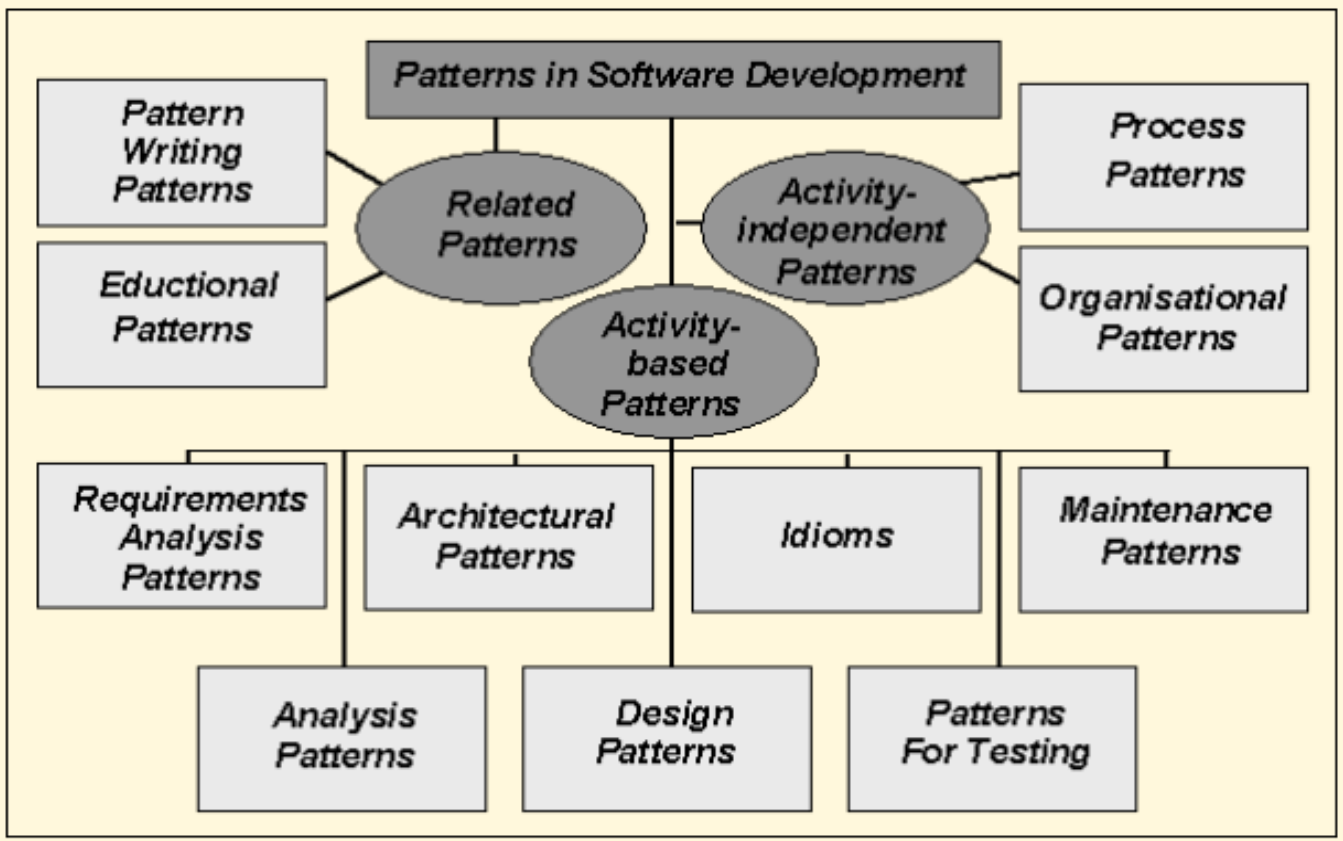
- шаблони за проектиране **(software design)**
  - Архитектурни **(системно проектиране)**
  - Дизайн **(микро-архитектури) [Gamma-GoF]**
  - Идиоми **(ниско ниво)**
- шаблони за анализ **(recurring & reusable analysis models) [Flower]**
- организационни шаблони **(структура на организация/проект)**
- процесни шаблони **(software process design)**
- домейн-специфични шаблони

стратегии от високо ниво, които засягат мащабни компоненти и глобалните свойства и механизми

парадигми и специфични за отделните езици програмни техники

Възможна класификация на шаблоните

- *Идиоми* = шаблони за имплемен-тация



***Източник*: Pattern-based Software Development An Empirical Study, by Thoralf Czichy, http://www.emailhub.org/patternsurvey/summary/index.htm**

Състав

Pattern

- **Context**
  - a design situation giving rise to a design problem
- **Problem**
  - a set of **forces** (goals and constraints) occuring in that context
- **Solution**
  - a form or rule that can be applied to **resolve** these forces

Пример – място до прозореца

- сили
  - **той/тя иска да седи удобно**
  - **той/тя обича светлината и да наблюдава външната среда**
- решение
  - **във всяка стая, най-малко един прозорец да има «място до прозореца»**

Форма на Александър (канонична форма) (1/2)

Име

Смислено име

Проблем

Изложение на проблема

Контекст

Ситуацията, породила проблема (пред-условия за повторение на проблема и решението му)

Сили

Описание на релевантни сили (цели) и ограничения

Решение

Изпитано решение на проблема

Примери

Примерни приложения на шаблона

Форма на Александър (канонична форма) (2/2)

Резултатен контекст (force resolution)
състояние на системата след прилагане на шаблона

Обосновка (Rationale)
обяснение на стъпките или правила в шаблона

Свързани шаблони
статични и динамични отношения между шаблоните

Известна употреба
появяване на шаблона и/или прилагането му в рамките на съществуваща система

GoF формат (1/2)

Име на шаблон и място в класификацията

Смисъл (Intent)

Какво прави шаблонът / кога решението работи

Познат още като

Други известни имена на шаблон (ако съществуват)

Мотивация

Проблем на дизайна/ как класовите и обектните структури решават проблема

Приложимост

Ситуации, където шаблонът може да се прилага

Структура

Графично представяне на класовете в шаблона

GoF формат (2/2)

Участници

Участващи класове/обекти и техните отговорности

Кооперирane

Как участващите класове/обекти решават съвместно проблема

Последици

Съображения, компромиси

Имплементация

Указания, техники

Примерен код

Фрагменти код с примерна имплементация

Известни използвания

Шаблони в реални системи

Свързани шаблони

Шаблони, отнасящи се към разглеждания шаблон

Формат (template) за описание на шаблон

[PATTERN-Име]
Author
[YOUR-Име] ([YOU@YOUR.ADDR]).
Last updated on [TODAY'S-DATE]
контекст
[PARAG-1]
[PARAG-2]
Проблем
[ONE-ASPECT]
[ANOTHER-ASPEC
Examples
сили
1.[FORCE-1]
2.[FORCE-2]
Design
[PARAG-1]
[PARAG-2]
An Implementation
[SOME-CODE]
Examples
Variants
[VARIANT]
[ANOTHER-VARIANT]
See Also
[ANOTHER-REF]

```
IF   you find yourself in контекст
                for example EXAMPLES,
                with Проблем,
                entailing сили
           THEN for some REASONS,
                apply DESIGN FORM AND/OR RULE
                to construct SOLUTION
                leading to NEW контекст and OTHER шаблони
```

http://g.oswego.edu/dl/pd-FAQ/pd-FAQ.html

http://hillside.net/шаблони/Writing/Lea.html

Други формати за шаблони:

- http://hillside.net/шаблони/template.html
- http://www.paterndepot.com/pages (Templates)

Шаблонен език **[Coplien]**

- Един шаблонен език определя *набор от шаблони и правила* за съчетаването им в *архитектурен стил*, или…
- …е структурирана колекция от шаблони, които се изграждат, за да трансформират нуждите и ограниченията в софтуерна архитектура **[Software Design Patterns: Common Questions and Answers]**
- …описва софтуерни рамки или семейства от свързани системи от шаблони **[Patterns Home Page -> Patterns Definitions]**

Каталог и система от шаблони

[Buschmann, the POSA (Pattern-Oriented-Software-Architecture) book]

- Каталог от шаблони

  …набор от свързани помежду си шаблони, които са разделени на малък брой категории…

- Система от шаблони

  …единен набор от свързани шаблони, които работят заедно в подкрепа на изграждането и еволюцията на цялата архитектура ...

Принципи на проектирането с шаблони [4] (1/3)

- **Principle Of Least Astonishment** (POLA) - когато два елемента на интерфейса са в конфликт или са двусмислени, поведението трябва да бъде това, което най-малко ще изненада потребителя, когато възникне конфликтът
- **Да правим обикновени неща лесни, а редките неща възможни**
- **Консистентност** – колкото повече произволни правила се натрупват върху програмиста (правила, които нямат нищо общо с решаването на проблема), толкова по-бавно програмистът може да ги кодира. И това не изглежда да е линейна, а експоненциална зависимост.

Принципи на проектирането с шаблони [4] (2/3)

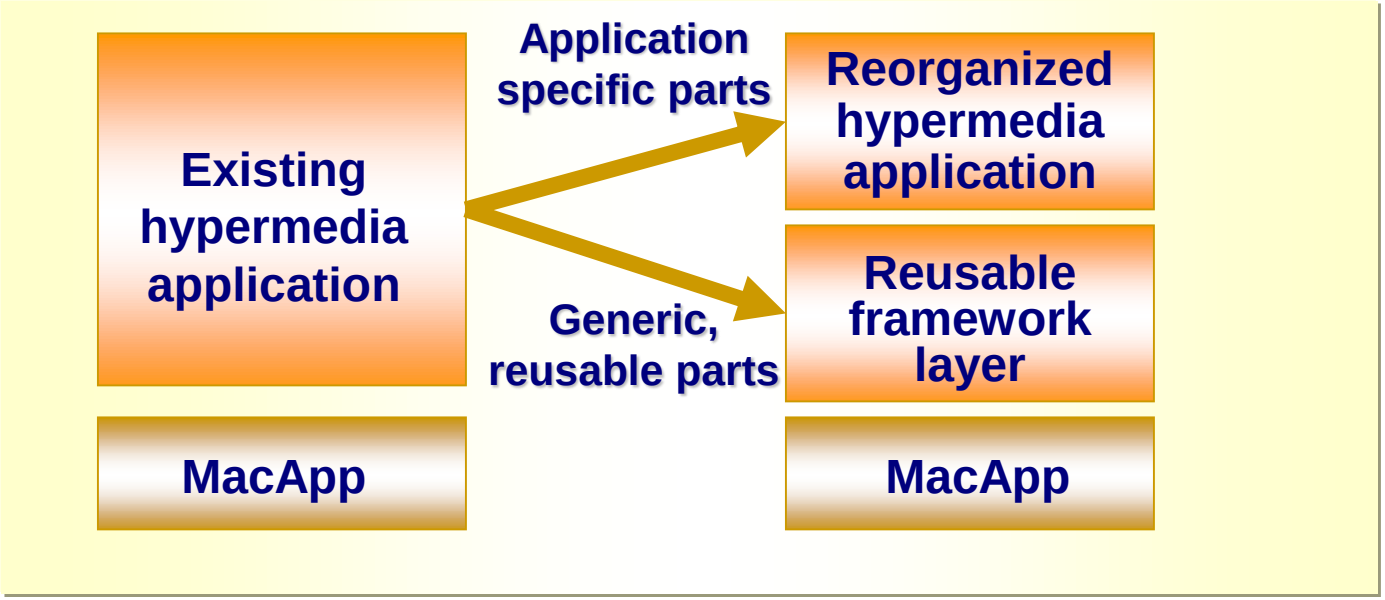
- **Закон на Demeter (Principle of Least Knowledge)**: "*Don't talk to strangers*." Един обект трябва да реферира само себе си, атрибутите си, както и аргументите на своите методи:
  - Всяка единица трябва да има само ограничени познания за другите единици – и то само за звената, "тясно" свързани с текущата единица.
  - Всеки блок трябва да говори само със своите приятели, но не и с непознати.
- **Изваждане (Subtraction)**: дизайнът е завършен, когато не можеш да изхвърлиш нищо от него

Принципи на проектирането с шаблони [4] (3/3)

- **Простота преди универсалност** - "най-простото работещо решение е най-доброто", най-добрия път за постигане на всеобщо решение е чрез разбирането на добре дефинирани конкретни примери.
- **Рефлексивност** - една абстракция в един клас, един клас на абстракция. Може също така да се нарича **изоморфизъм.**
- **Независимост или *ортогоналност*** – изразяване на независими идеи самостоятелно. Това допълва разделянето, капсулирането и вариацията, и е част от принципа *ниска свързаност - висока кохезия*
- **Веднъж и само веднъж** - да се избегне дублирането на логиката и структурата, където дублиране не е случайно, т.е. когато два участъка код изразяват едно и също намерение (цел).

Реорганизация чрез шаблони

- Виж “Experiences using Design Patterns to Reorganize an Object-Oriented Application”, от Walter Zimmer
- Цел на реорганизацията - да отстрани недостатъците в проектирането и изпълнението; по-добра поддръжка



Стъпки в реорганизацията (1/2)

Предварителни стъпки:

1 документация

2 откриване на изходни точки

- **идентифициране на класове/ подсистеми с цели, подобни на тези на шаблоните за проектиране**
- **опит и бъдещи сценарии**
- **метрики и правила за дизайн**
- **анализ на приложението за съществуващи шаблони**

шаблони за проектиране - части от системата

критични части на приложението и техните зависимости

Стъпки в реорганизацията (2/2)

Същинска реорганизация:

1 намиране и изследване на подходящи шаблони за проектиранes

2 реконструкция и документация

- **приложни класове, съответстващи на шаблон за проектиране**
- **включване на имената на класовете на приложението в шаблона за проектиране (напр. LinkStrategy, HyperMediaDecorator)**

Резултати и познания

Резултати

- драстично намаляване на зависимостите между подсистеми
- кратка проектна документация

Познания

- общ речник – основно предимство
- реорганизацията е интензивна по време задача
- добро познаване на необходимите шаблони за проектиране
- комбинация от няколко шаблона

Процес на изучаване [4]

Няколко стъпки в учебния процес за шаблони за проектиране:

- 1. **Приемане (Acceptance)** - приемаме предпоставката, че шаблоните за проектиране са важни за работата ни.
- 2. **Признаване (Recognition)** - признаваме, че трябва да прочетем за шаблоните за проектиране повече, за да знаем кога може да ги използваме.
- 3. **Интернализация (Internalization)** - вътрешно приемане на шаблоните достатъчно подробно, че да знаем кои от тях и как могат да ни помогнат да решим даден проблем.

Три популярни групи шаблони за проектиране

Каталогът от базови шаблони за проектиране включва **23** общи полезни шаблона за конструиране на ОО софтуер, разделени в 3 групи:

- *Създаващи шаблони* - тези, които създават обекти вместо нас, т.е. вместо ние да инстанцираме обекти директно. Това дава възможност за по-голяма гъвкавост при вземането на решение кои обекти трябва да бъдат създадени за даден случай.
- *Структурни шаблони* - помагат ни да композираме групи от обекти в по-големи структури, като например сложни потребителски интерфейси или счетоводни данни.
- *Поведенчески шаблони - за* определяне комуникацията между обекти в дадена система и как се контролира потока управление в комплексна програма.

Класификация на шаблони за проектиране

По обхват – дали шаблонът използва предимно класове или обекти:

- Шаблоните, ориентирани към класове, работят с наследяване и референтни връзки между класове и подкласове. Повечето връзки са установени чрез наследяване и оттам са статични.
- Шаблоните, ориентирани към обекти, работят с отношения между обекти, които могат да се променят динамично по време на изпълнение на програмата

Класификация на шаблоните за проектиране

По предназначение – какво прави шаблонът

- Създаващи шаблони
  - Резюмират процеса на инстанциране на обекти
  - Правят използваната система по-малко зависима от нейната реализация
  - Създаващите шаблони, ориентирани към класове, използват наследяване, за да променят инстанция на класа
  - Създаващите шаблони, ориентирани към обекти, делегират инстанцирането към друг обект

Класификация на шаблони за проектиране

- Структурни шаблони - за формиране на големи структури от данни
  - структурните шаблони , ориентирани към класове използват агрегацията от класове
  - структурните шаблони, ориентирани към обекти, използват агрегацията от обекти

Класификация на шаблони за проектиране

- Поведенчески шаблони
  - Касаят алгоритми и присвояване и управление на отговорности между обекти
  - Описват шаблони за комуникация между класове и обекти
  - Поведенческите шаблони, ориентирани към класове, използват наследяване, за да разпределят поведението между класовете
  - Поведенческите шаблони, ориентирани към обекти, използват композиции от обекти, за да разпределят поведението между класовете

Шаблони за проектиране, каталог GoF – оригинални имена

		Цел		
		Създаващи	**Структурни**	**Поведенчески**
Обхват	**Клас-базирани**	•**Factory Method**	•**Adapter**	•**Interperter**
	Обектно-базирани	•**Abstract Factory** •**Builder** •**Prototype** •**Singleton**	•**Bridge** •**Composite** •**Decorator** •**Facade** •**Flyweight** •**Proxy**	•**Chain of Responsibility** •**Command** •**Iterator** •**Mediator** •**Template method** •**Memento** •**Observer** •**State** •**Strategy** •**Visitor**

Шаблони за проектиране, каталог GoF – превод на български език

		Цел		
		Създаващи	**Структурни**	**Поведенчески**
Обхват	**Клас-базирани**	•Метод фабрика	•Адаптер	•Интерпретатор
	Обектно-базирани	•Абстрактна фабрика •Строител •Прототип •Сек	•Мост •Композиция •Декоратор •Фасада •Мини-обект •Пълномощник	•Верига от отговорности •Команда •Итератор •Посредник •Шаблонен метод •Спомен •Наблюдател •Състояние •Стратегия •Посетител

За самостоятелна работа – вижте статията

- **Non-Software Examples of Software Design Patterns,** от Michael Duell, в AG Communication Systems e-zine, 29/03/2007