

Padrões de Desenho

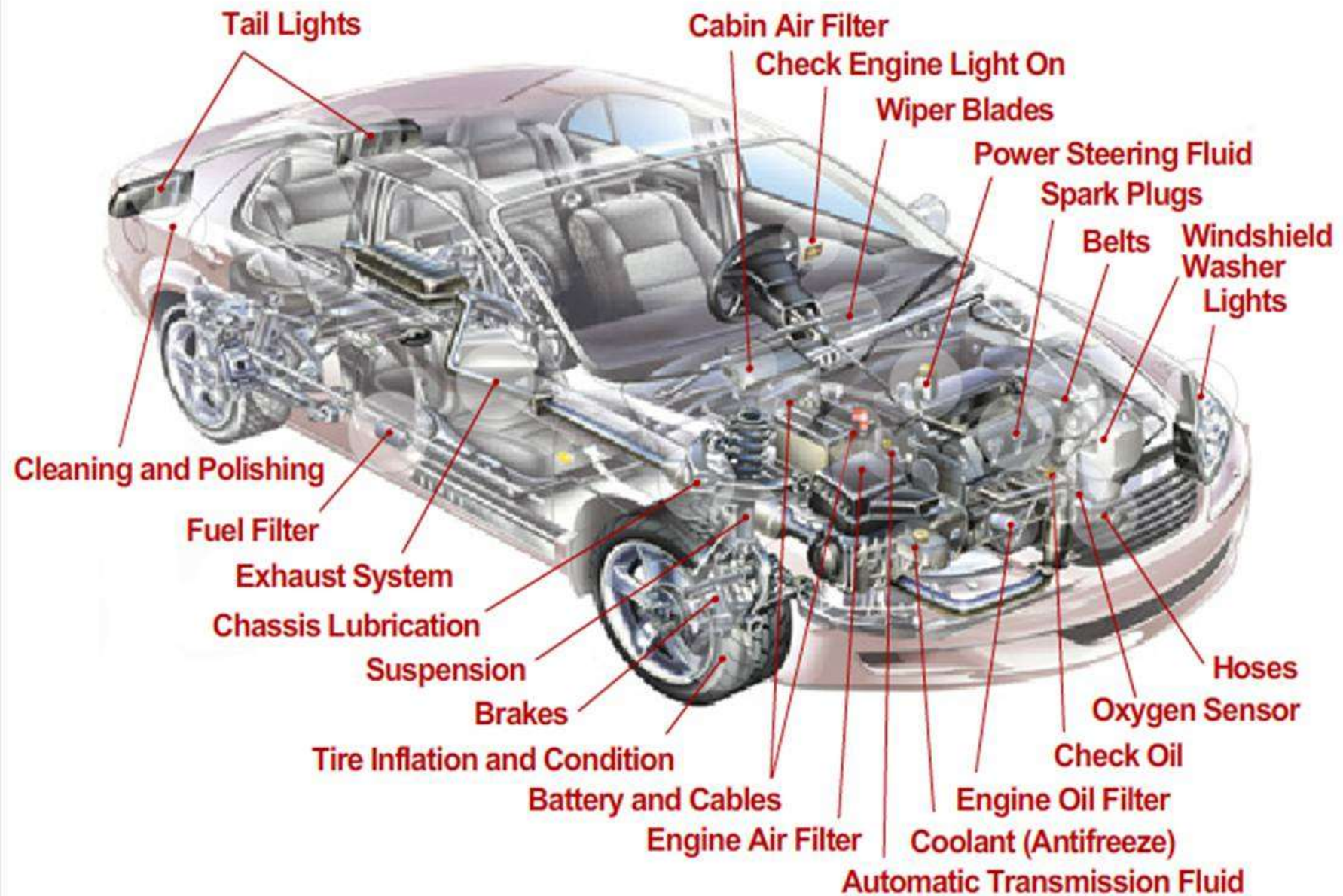
workshop

Agenda

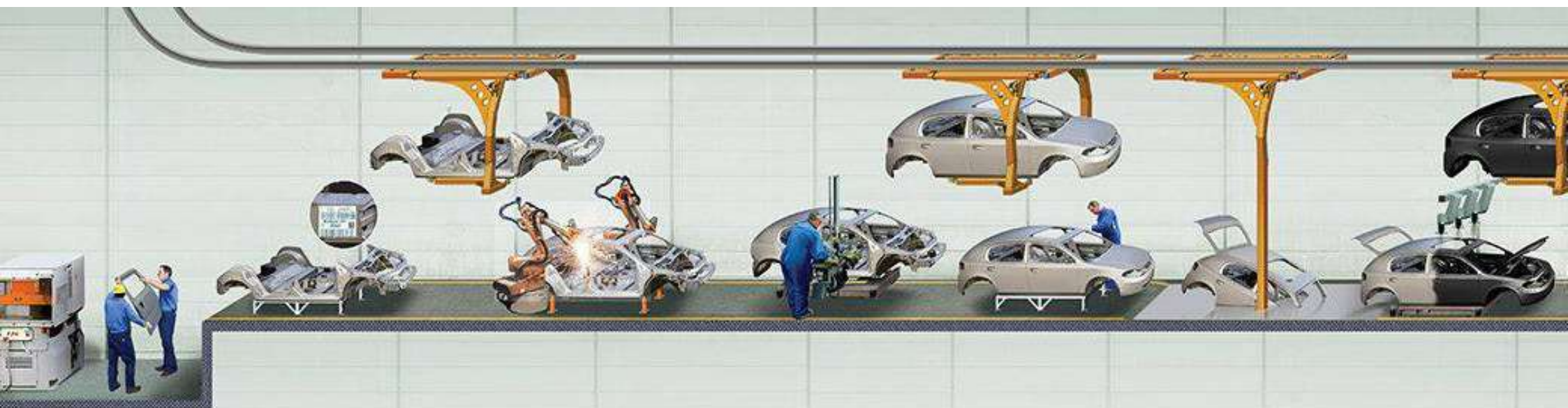
- Introdução
- Definição
- Implementação
- GoF

Introdução

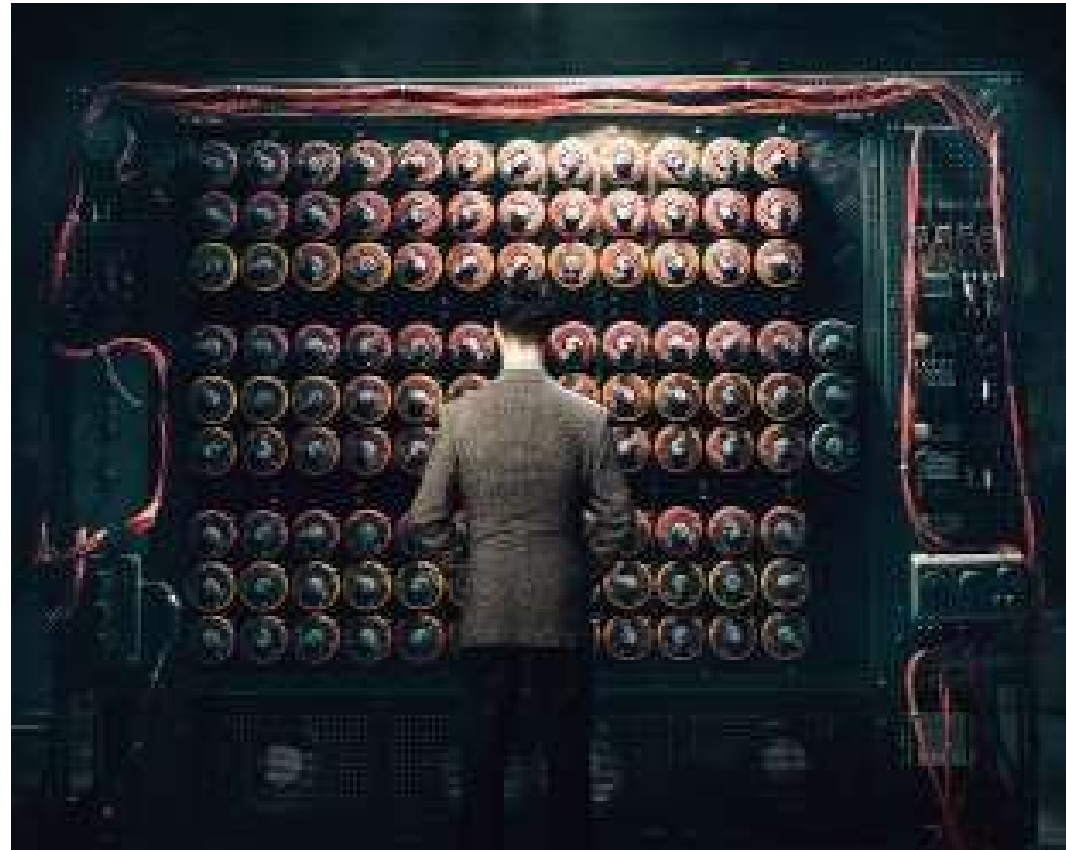




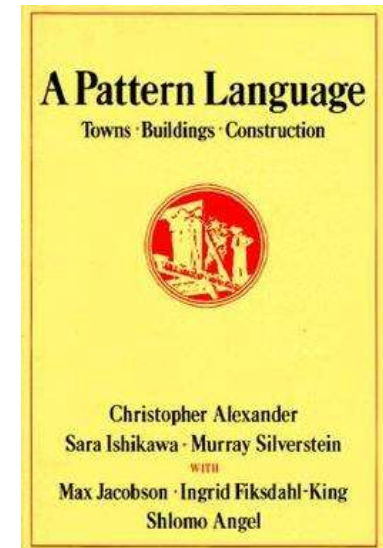




Definição



Definição



Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice

Definição



- Serve como identificador
- O nome deverá representar o problema, a solução e suas consequências.



- O problema descreve quando aplicar o padrão
- O contexto da sua aplicação
- Lista de pré-condições que se têm de verificar para que seja viável a utilização do padrão
- Cenário exemplo de aplicabilidade a exprimir o problema a resolver
- Partes que constituem o padrão

Definição

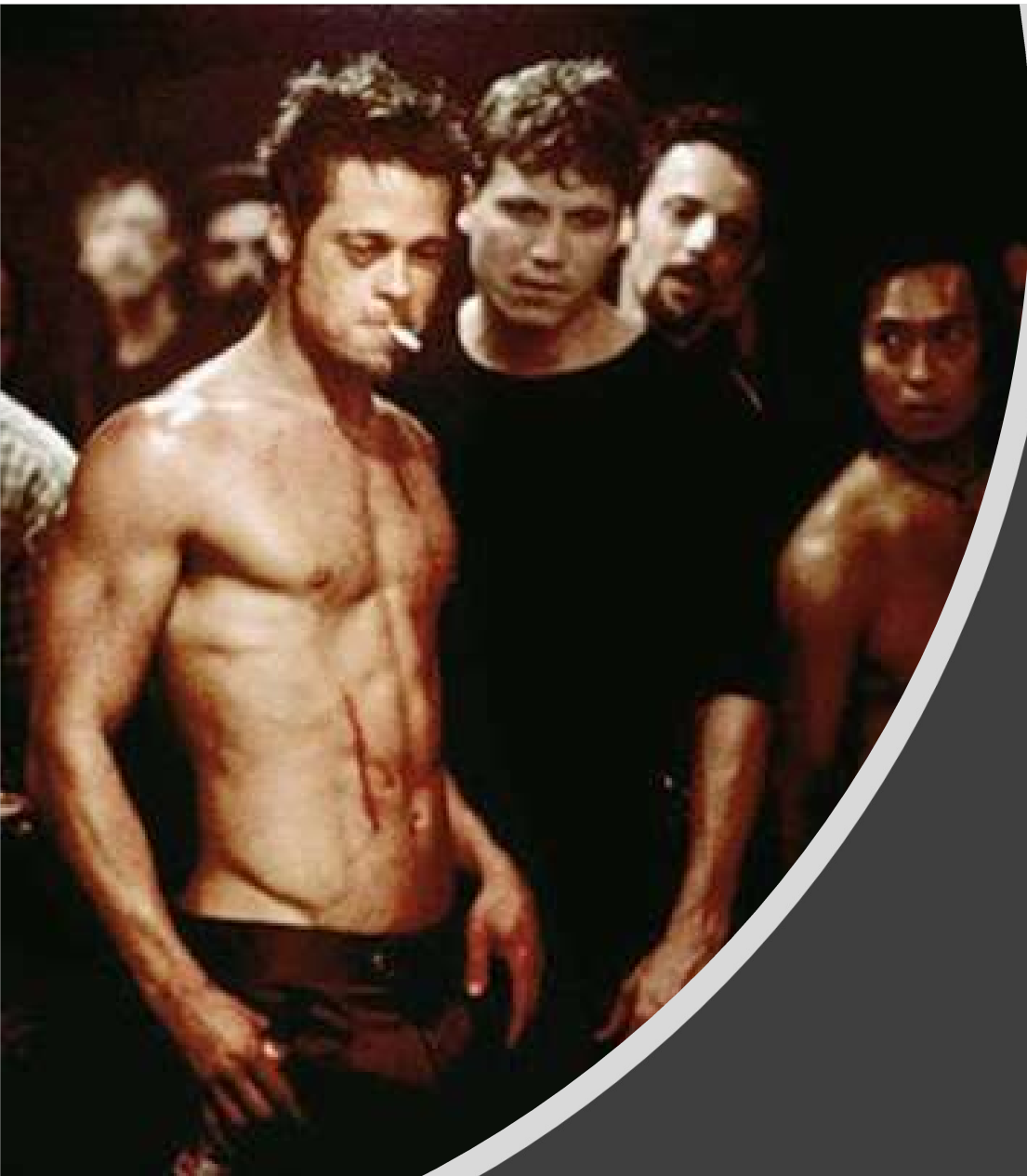


A solução é a definição de um template que descreve:

- Composição de partes
- Relação entre elas
- Colaboração entre as partes



- Consequências são resultado de escolhas
- Servem para avaliar custo vs benefício
- Como principais *drivers* considere-se:
 - Flexibilidade
 - Extensabilidade
 - Portabilidade
 - *Trade-offs* espaço vs tempo



Implementação

Implementação

Considerações

- Objectos relevantes ao domínio do problema
- Granularidade dos objectos
- Especificar interfaces dos objectos
- Especificar implementação dos objectos
 - Classe vs interface
 - Interações com a interface não com a a sua implementação
- Reutilização
 - Herança vs Composição
 - Delegação
 - Tipos parameterizaveis

Implementação

Mudança & Impactos

- Evitar a instanciação de objectos explicita
- Evitar a realização de tarefas explicita
- Evitar dependências para plataforma & sistema operativo
- Abstrair a representação e implementação concreta
- Reduzir a dependência directa de algoritmos
- Evitar acoplamento entre classes
- Evitar herança para extender funcionalidades

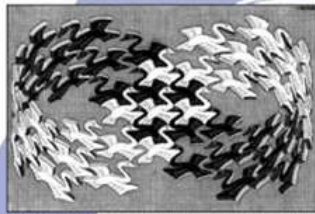
GoF



Design Patterns

Elements of Reusable
Object-Oriented Software

Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides



Foreword by Grady Booch

- Criação
 - Como são criados os objectos?
- Estruturais
 - Como organizar classes para formar novas estruturas de dados
- Comportamentais
 - Associação de responsabilidades a classes e como comunicam entre si

		Purpose		
		Creational	Structural	Behavioral
Scope	Class	Factory Method	Adapter	Interpreter Template Method
	Object	Abstract Factory Builder Prototype Singleton	Adapter Bridge Composite Decorator Facade Proxy	Chain of Responsibility Command Iterator Mediator Memento Flyweight (195) Observer State Strategy Visitor

