Conceitos de Programação Orientada a Objetos

Eduardo Figueiredo

http://www.dcc.ufmg.br/~figueiredo

Visão Geral

- Um programa OO é geralmente constituído de várias classes
 - Cada classe possui vários métodos (comportamento)
 - Classes também possuem atributos (estados)
- Classes trocam mensagens entre si
 - Chamada de métodos

Analogia: Carro

- Funcionalidades
 - Andar mais rápido: acelerar
 - Andar mais devagar: frear
- Atributos
 - Constantes: Cor, número de portas, capacidade do tanque, ...
 - Dinâmicos: Quantidade atual de combustível, velocidade atual, ...

Ocultando os Detalhes

- Os pedais freio e acelerador "ocultam" do motorista os reais mecanismos que fazem o carro andar mais rápido ou mais devagar
- Pessoas com pouco ou nenhum conhecimento de mecânica / hidráulica podem dirigir um carro



Envio de Mensagens

Acelerar

 O pedal acelerador envia uma mensagem para o carro aumentar a velocidade

Frear

 O pedal de freio envia uma mensagem para o carro diminuir a velocidade



Conhecendo a Si Próprio

- Um carro sabe a sua velocidade atual
 - Mas, não sabe a velocidade de outros carros



Minha velocidade atual é 60 Km/h

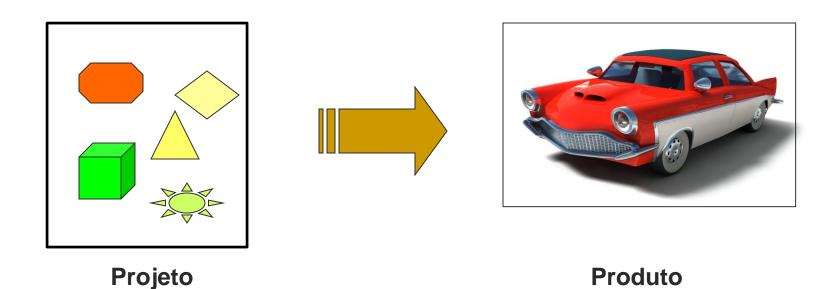


Implementando um Projeto

- Assim como em outras engenharias, o projeto não é suficiente
 - Ninguém pode dirigir o desenho de um carro
 - Ninguém pode executar o projeto de um sistema
 - Por enquanto...

Do Projeto para o Produto

 Antes do carro ficar pronto, ele deve ser projetado



Do Projeto para Implementação

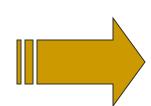
 Antes do carro ser codificado, ele deve ser projetado

Carro

cor

velocidadeAtual

acelerar() frear()



```
public class Carro {
   String cor;
   int velocidadeAtual;

   void acelerar() {}
   void frear() {}
}
```

Projeto

Implementação

A Tecnologia Java

Evolução das Linguagens

- Linguagens de Máquina
 - +1300042774 (traduzido depois para 0 e 1)
- Linguagens Assembly
 - load basepay / add overpay / store grosspay
- Linguagens de Alto Nível
 - o grosspay = basepay + overpay

De onde veio Java?

BCLP (1967) por Martin Richards



Linguagem C (1972) por Dennis Ritchie



C++ (1980) por Bjarne Stroustrup, Bell



Java (1995) pela Sun

Bibliotecas de Java (API)

- Organização da Biblioteca
 - Pacotes -> Classes -> Métodos (função)

- Por que usar classes da biblioteca?
 - Ganhar tempo
 - Mais confiáveis
 - São portáveis
 - São eficientes, etc.

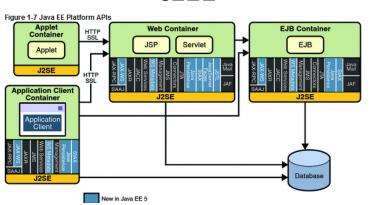
Plataforma Java

Principais Edições

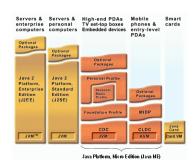
J2SE



J2EE



J2ME



J2SE (Java Padrão)

	Java Language		Java Language													
JDK	Tools & Tool APIs		java	javac	avac javadoc		jar	j	avap	JPDA		jconsole				
			Security	Int'l	RMI	IDL	Deploy	Moi	nitoring	Troubl	eshoot	ot Scriptin		JVM TI		
		Deployment Technologies	Dej	ployme	nt		Java	Web	Start	Java Plug-in						
		User Interface Toolkits	AWT					S	wing	Java 2D)		
			Accessibility Drag n			ор	Input Meth		s Im	Image I/O		Print Service		Sound		
		Integration Libraries	IDL	JD	JDBC™		JNDI™		MI RMI-IIO		P Scr		ripting			
	JRE	Other Dase	Beans Intl S		Intl Suppo	tl Support		JMX		JNI				Math		
		Libraries	Networking	Ove	Override Mechani		Security Serializ		erialization	tion Extension I		lechanism		XML JAXP		
		lang and util Base Libraries	lang and util		Collections		oncurrency Utilities		JAR		Logging		Ma	Management		
		Base Libraries	Preferences API		Ref Objects		Reflection		Regular Expressions		Versioning Zip		Zip	Instrument		
		Java Virtual Machine		Java Hotspot _™ Client VM						Java Hotspot _™ Server VM						
		Platforms	Solaris [™]			Linux			Windows				Other			

Java SE API

Bibliografia

- DEITEL, H. M.; DEITEL P. J. Java: Como Programar, 8a. Edição. Pearson, 2010.
 - Seções 1.5 a 1.10
 - Capítulo 3
- BOOCH, G., RUMBAUGH, J.,
 JACOBSON, I. UML, Guia do Usuário.
 Rio de Janeiro: Campus, 2000.