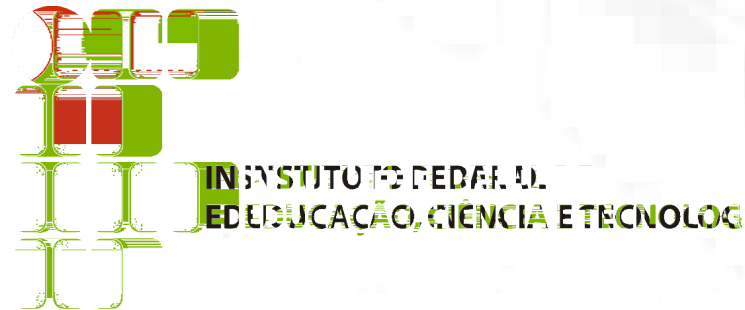


INF011 – Padrões de Projeto

00 - Apresentação

Sandro Santos Andrade
sandroandrade@ifba.edu.br

Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Departamento de Tecnologia Eletro-Eletrônica
Graduação Tecnológica em Análise e Desenvolvimento de Sistemas



Roteiro

- Contexto e motivação
- Padrões de Projeto
- Principais de Projeto \$ \$
- Estudo de Caso & *Bridge*
- Padrões Arquiteturais e Idio (as
- Conclusão

Contexto e Motivação

- * (a r+pida est , ria&
- - o" . / u (professor "o (a responsabi)idade de infor (ar aos estudantes0 ao fina) da sua au)a0 o)o"a) onde o"orrer+ a au)a se1uinte
- * (a pri (eira i (p)e (entação&
 1. \$bten2a a)ista de estudantes da ")asse
 3. Para "ada estudante0 faça o se1uinte&
 1. %n"ontre a pr , xi (a au)a 'ue e)e assistir+
 3. 4o"a)i5e a sa)a onde o"orrer+ esta au)a
 6. %n"ontre u ("a (in2o da sua sa)a para a sa)a)o"a)i5ada
 7. Infor (e ao estudante o "a (in2o a ser se1uido

Contexto e Motivação

- * (a r+pida est , ria&
- - o" . / u (professor "o (a responsabi)idade de infor (ar aos estudantes0 ao fina) da sua au)a0 o)o"a) onde o"orrer+ a au)a se1uinte
- * (a pri (eira i (p)e (entação&
 - Programa de Controle {
 1. \$bten2a a)ista de estudantes da ")asse
 3. Para "ada estudante0 faça o se1uinte&
 1. %n"ontre a pr , xi (a au)a 'ue e)e assistir+
 3. 4o"a)i5e a sa)a onde o"orrer+ esta au)a
 6. %n"ontre u ("a (in2o da sua sa)a para a sa)a)o"a)i5ada
 7. Infor (e ao estudante o "a (in2o a ser se1uido

Contexto e Motivação

- * (a r+pida est, ria&
- - o" . / u (professor "o (a responsabi)idade de infor (ar aos estudantes0 ao fina) da sua au)a0 o)o"a) onde o"orrer+ a au)a se1uinte
- * (a outra i (p)e (entação&
 1. Pub)i ' ue no (ura) os "a (in2os da sua sa)a para todas as outras
 3. Infor (e para toda a tur (a& 8\$ (ura) "ont/ (os "a (in2os para as outras sa)as. * ti)i5e (-o e dirija (-se para as suas pr , xi (as au)as9

Contexto e Motivação

- \$ 'ue (udou entre as so)uções :
 - Na pri (eira0 !o" . presta atenção a u (a s/rie de deta)2es e / respons+!e) por tudo
 - Na se1unda0 instruções b+si"as são forne"idas e espera-se 'ue "ada pessoa saiba fa5er o seu traba)2o

Contexto e Motivação

- \$ 'ue (udou entre as so)uções :
 - Na pri (eira0 !o" . presta atenção a u (a s/rie de deta)2es e / respons+!e) por tudo
 - Na se1unda0 instruções b+si"as são forne"idas e espera-se 'ue "ada pessoa saiba fa5er o seu traba)2o

DESVIO DE RESPONSABILIDADES

Contexto e Motivação

- \$ 'ue (udou entre as so)uções :
 - Na pri (eira0 !o" . presta atenção a u (a s/rie de deta)2es e / respons+!e) por tudo
 - Na se1unda0 instruções b+si"as são forne"idas e espera-se 'ue "ada pessoa saiba fa5er o seu traba)2o

DESVIO DE RESPONSABILIDADES

ESTUDANTES RESPONSÁVEIS PELO SEU PRÓPRIO COMPORTAMENTO

Contexto e Motivação

- Por 'ue esta reorganização de responsabilidades / i (portante :
 - ; upon2a 'ue a)unos bo)sistas ten2a (de preen"2er u (for (u)+rio de a!a)iação da au)a antes de se diri1ir < au)a se1uinte
 - Na pri (eira so)ução o pro1ra (a de "ontro)e de!e ser (odifi"ado de (odo a distin1uir os tipos de a)unos e dar instruções espe"#fi"as a "ada tipo
 - Na se1unda so)ução o "o (porta (ento dos a)unos bo)sistas seria (odifi"ado e o pro1ra (a de "ontro)e "ontinuaría di5endo 8 - + para a sua pr , xi (a au)a9

Contexto e Motivação

- Por 'ue esta reor1ani5ação de responsabilidades / i (portante :
 - ; upon2a 'ue a)unos bo)sistas ter2a (cl)preen"2er u (for (u)+rio de a!a)iação da au)a antes de se diri1ir < au)a seguinte
 - Na pri (eira so)uçã (o)pro1ra (a de "ontro)e de!e ser (odifi"ado de (odifi)distin1uir os tipos de a)unos e dar instruções espe1ais a "ada tipo
 - Na se1unda so)uçã o "o (porta (ento dos a)unos bo)sistas seria (odifi"ado e o pro1ra (a de "ontro)e "ontinuar di5endo 8 - + para a sua pr , xi (a au)a9

Contexto e Motivação

-
- DESIGN FOR CHANGE

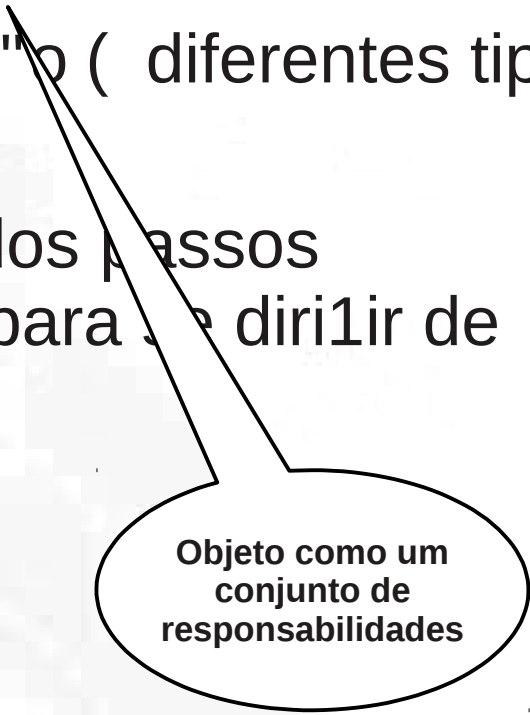
Quero saber mais !

Contexto e Motivação

- Para 'ue isso fun"ione / ne"ess+rio 'ue&
 - \$s a)unos "on2eça (o seu tipo =bo)sista > não bo)sista? e seja (respons+!eis pe)o seu pr ,prio "o (porta (ento
 - \$ pro1ra (a de "ontro)e se "o (uni'ue "o (diferentes tipos de a)unos de (aneira unifor (e
 - \$ pro1ra (a de "ontro)e não dependa dos passos parti"u)ares 'ue estudantes exe"uta (para se diri1ir de u (a sa)a < outra

Contexto e Motivação

- Para 'ue isso fun"ione / ne"ess+rio 'ue&
 - \$s a)unos "on2eça (o seu tipo =bo)sista > não bo)sista? e seja (respons+!eis pe)o seu pr ,prio "o (porta (ento
 - \$ pro1ra (a de "ontro)e se "o (uni'ue "o (diferentes tipos de a)unos de (aneira unifor (e
 - \$ pro1ra (a de "ontro)e não dependa dos passos parti"u)ares 'ue estudantes exe"uta (para se diri1ir de u (a sa)a < outra



Objeto como um conjunto de responsabilidades

Contexto e Motivação

- Para que isso funcione / necessário que
 - Os objetos tenham uma interface clara (o seu tipo é bom ou não é bom?) e seja responsável pelo seu próprio trabalho (porta-ento)
 - Os objetos devem ser tratados de maneira uniforme (de objetos de diferentes tipos)
 - Os objetos não dependam dos passos particulares que os estudantes executam para se dirigir de um lugar a outro

Interfaces como
um mecanismo
para representar
conceitos

Objeto como um
conjunto de
responsabilidades

Contexto e Motivação

- Para que isso funcione / necessário que
 - Os objetos tenham uma interface clara (o seu tipo é bom ou não é bom?) e seja responsável pelo seu próprio trabalho (porta-ento)
 - Proporcionar a interação entre os objetos (unidade de diferentes tipos de objetos de maneira uniforme)
 - Proporcionar a interação entre os objetos não dependa dos passos particulares que os estudantes executam para se dirigir de um lugar a outro

Encapsulamento
para prover vários
tipos de
ocultamentos

Interfaces como
um mecanismo
para representar
conceitos

Objeto como um
conjunto de
responsabilidades

Contexto e Motivação

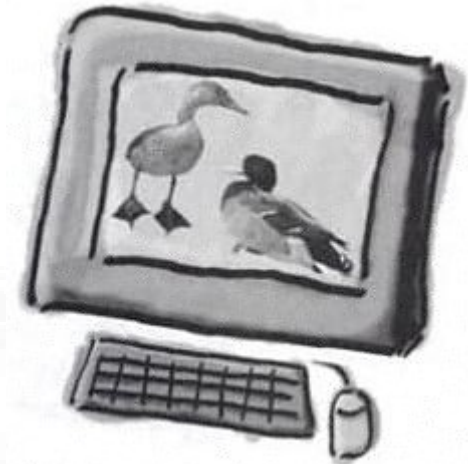
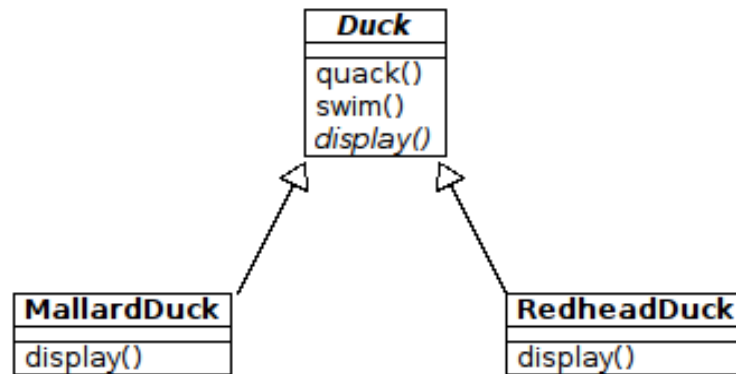
- Perspectivas no processo de desenvolvimento (então

PERSPECTIVA	DESCRIÇÃO
CONCEITUAL	Representa os conceitos no do (sistema) estudado Qual a minha responsabilidade ?
ESPECIFICAÇÃO	Foco nas interfaces não na implementação Como eu sou utilizado ?
IMPLEMENTAÇÃO	Foco na implementação Como eu cumpro minhas responsabilidades ?

FoA)er0 artin. * 4 Bisti))ed

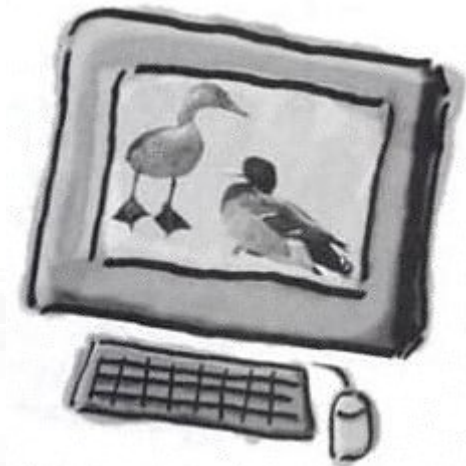
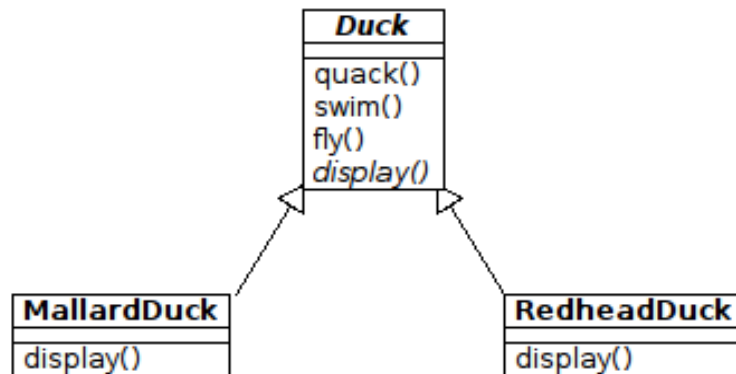
Contexto e Motivação

- Por 'ue pre"isa (os de padrões de projeto :
 - %xe (p)o& jo1o de patos



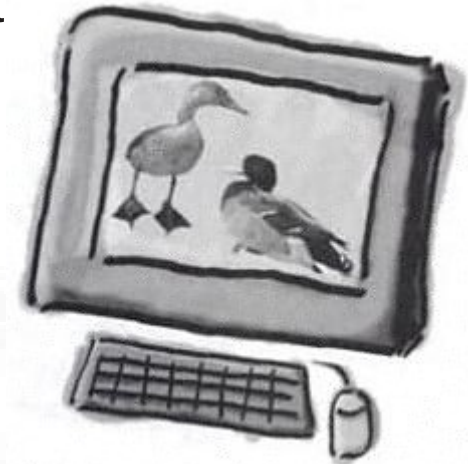
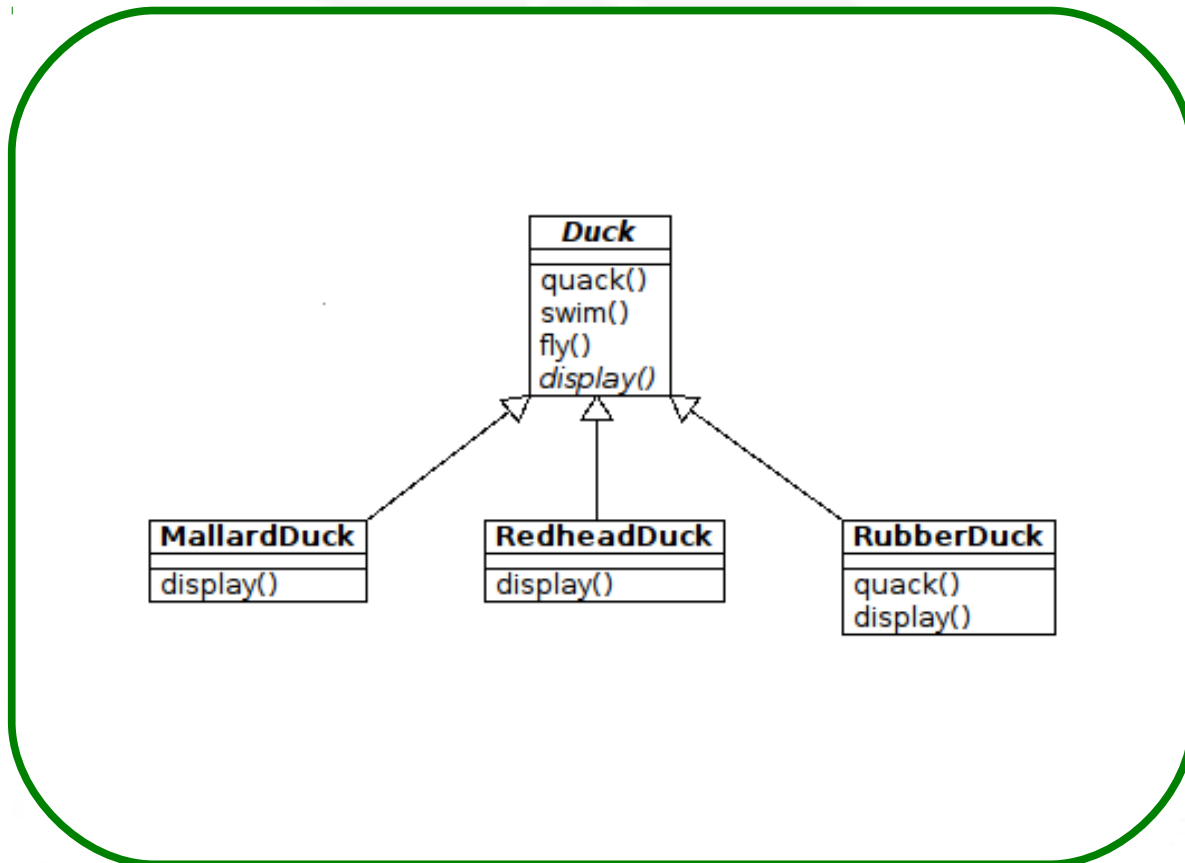
Contexto e Motivação

- Por 'ue pre"isa (os de padrões de projeto :
 - 1ª (udança& patos a1ora pode (!oar



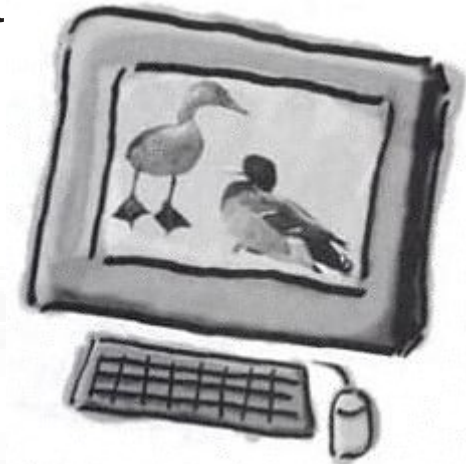
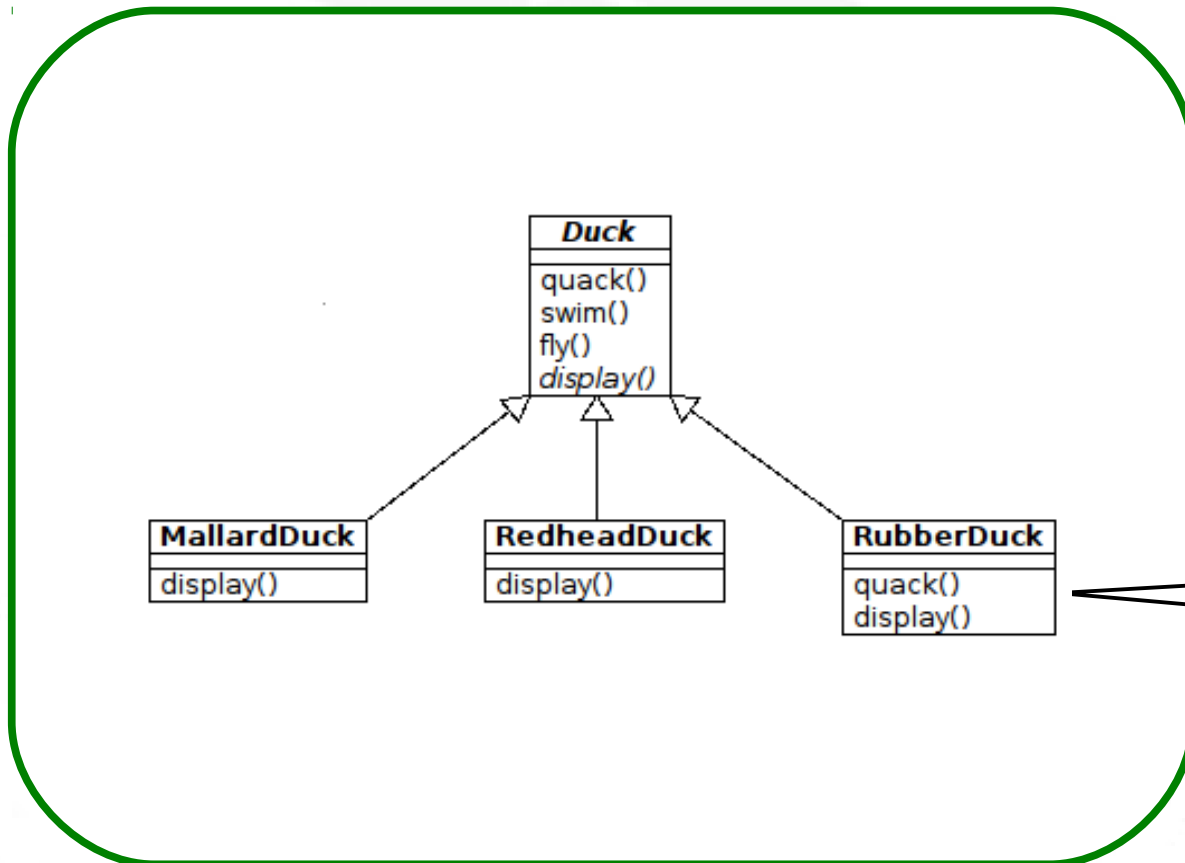
Contexto e Motivação

- Por 'ue pre"isa (os de padrões de projeto :
 - 3ª (udança& suportar patos de borra"2a



Contexto e Motivação

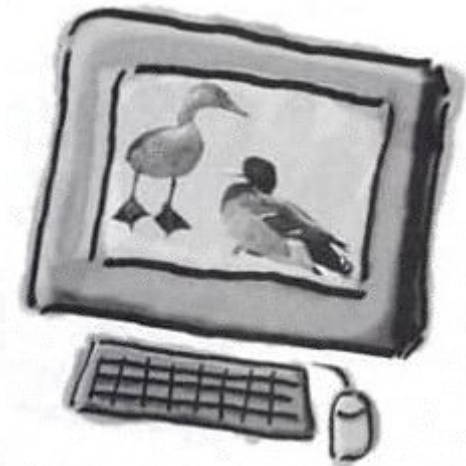
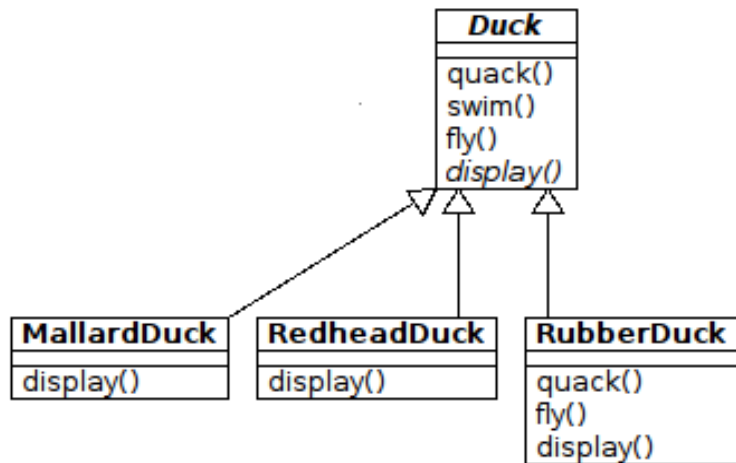
- Por 'ue pre"isa (os de padrões de projeto :
 - 3ª (udança& suportar patos de borra"2a



Hmm, patos de borracha não devem voar. O que fazer ?

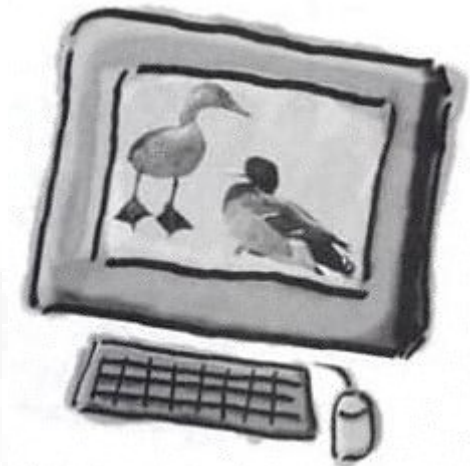
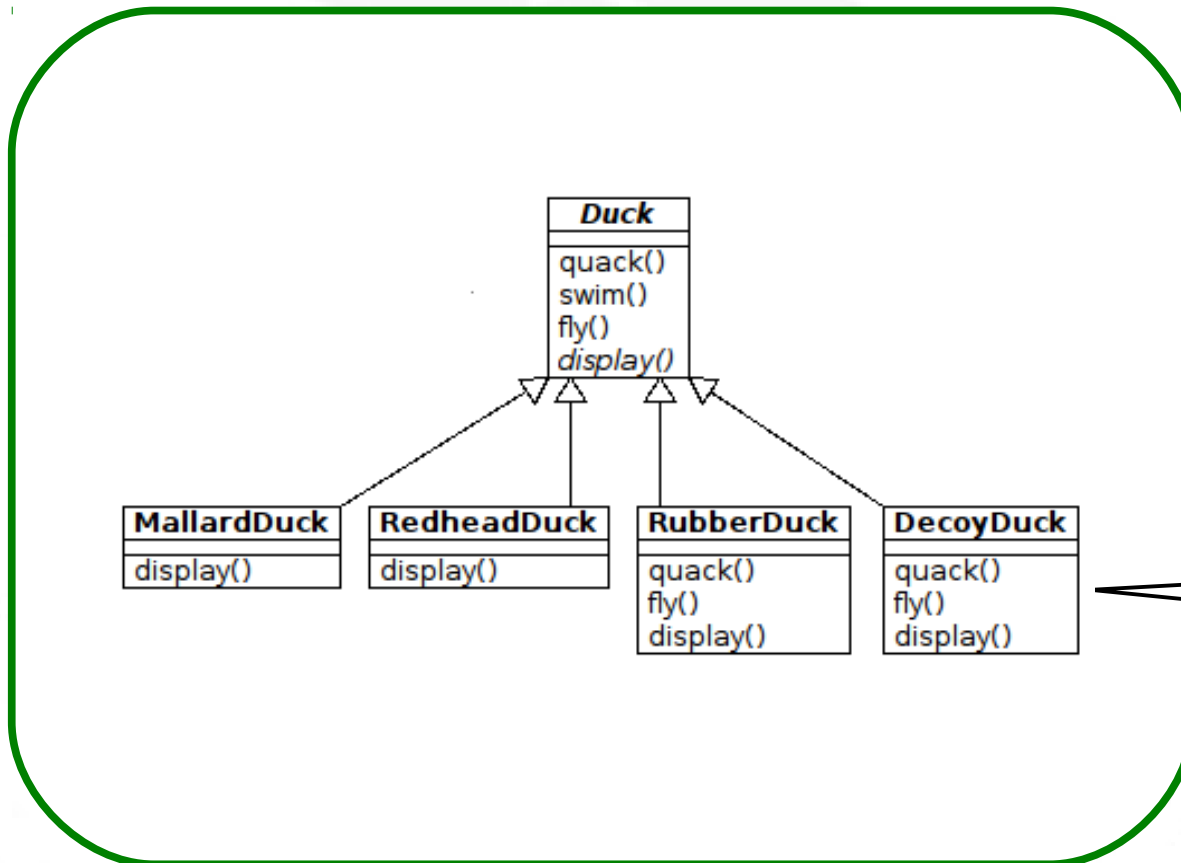
Contexto e Motivação

- Por que precisa (os) de padrões de projeto :
 - 1ª tentativa de sobreposições :asias :



Contexto e Motivação

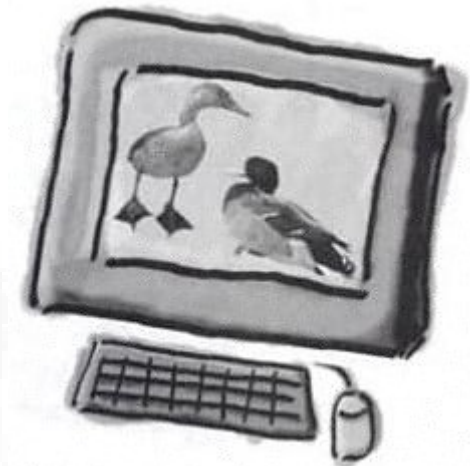
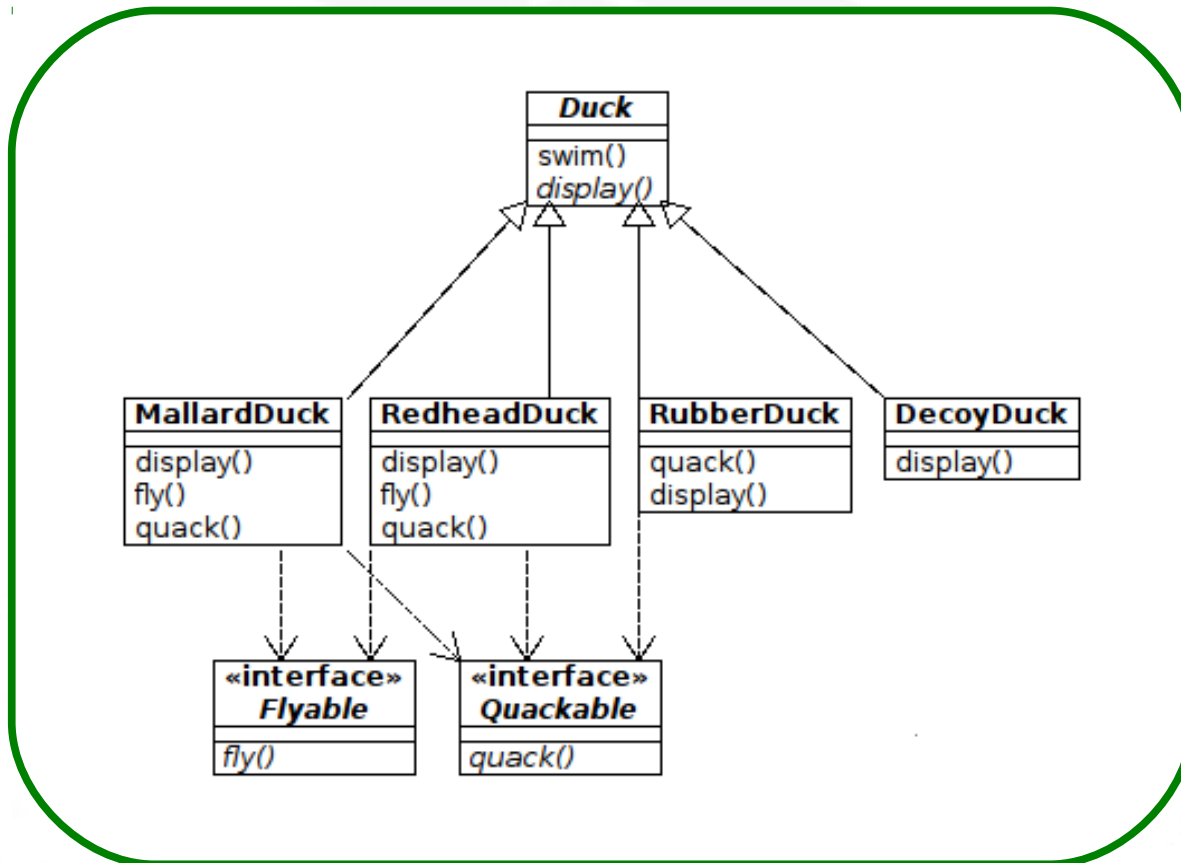
- Por que precisa (os de padrões de projeto):
 - 1ª tentativa & sobreposições & áreas:



E se tivéssemos
que suportar
patos decorativos
?

Contexto e Motivação

- Por que precisa (os) de padrões de projeto :
 - 3ª tentativa de interfaces :

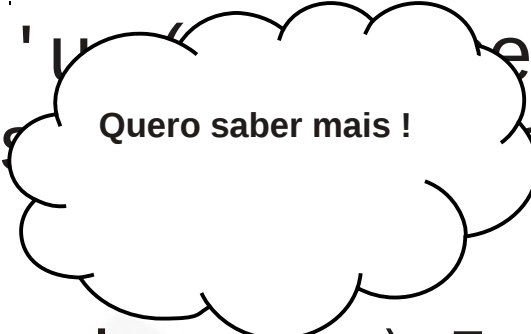


Padrões de Projeto

- A beleza está nos olhos de quem vê (! . ou pessoas "con"ordaria ('ue "ertas "oiasas são bonitas e outras não :9 [Christopher Alexander]
- Culturas te"e (ju)1a (entos sobre a beleza trans"edendo "renças individuais [Ruth Benedict]

Padrões de Projeto

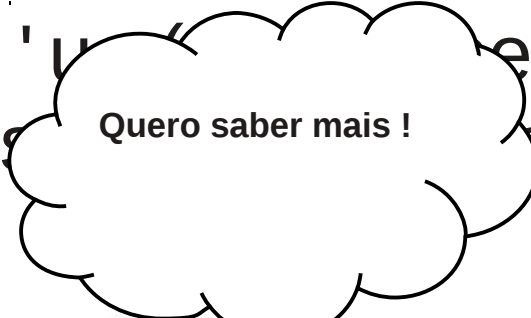
- A beleza está nos olhos de quem vê. Pessoas "ordinárias" que "certas" coisas não são. [Christopher Alexander]
- Culturas têm (juízo) sobre a beleza transcendendo diferenças individuais [Ruth Benedict]



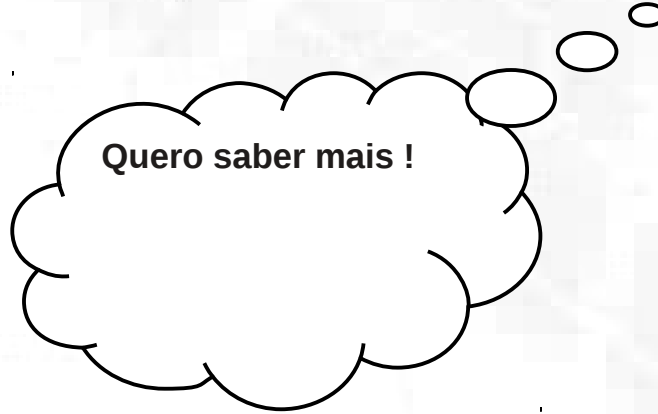
Quero saber mais !

Padrões de Projeto

- A beleza está nos olhos de quem vê. Pessoas diferentes veem coisas diferentes e outras não :9 [Christopher Alexander]
- Culturas te dão uma lente sobre a beleza transformando preferências individuais [Ruth Benedict]



Quero saber mais !



Quero saber mais !

Padrões de Projeto

- % o 'ue isso te (a !er "o (*software* :
 - uitos prob) e (as en"ontrados no desen!o)!i (ento de *software* são re"orrentes e pode (ser reso)!idos se1uindo u (a (es (a estrat/1ia =padrão de projeto?
 - Pode-se projetar *software* atra! /s da identifi"ação ante"ipada destes padrões
 - Padrões de projeto propor"iona (so)uções f)ex#!eis0 'ue a"o (oda ((udanças futuras
- *Design Patterns: Elements of Reusable Object-Oriented Software =Gang of Four - 1CC7?*

Padrões de Projeto

Padrões de Projeto são descrições de objetos e classes que se comunicam e que são configurados para resolver um problema genérico de projeto OO em um contexto particular

Page 1 of 4 - 1CC7

- Características
 - Promove o reuso de boas soluções
 - Estabelece uma terminologia comum
 - Antecipa discussões no desenvolvimento de projeto e evita mal-entendidos
 - São desobertos e não inalterados

Princípios de Projeto OO

- Os princípios de projeto OO são:
 1. Os objetos são definidos por um contrato de interface e este contrato não é violado
 2. Todos os dados são privados se exceções
 3. Deve ser possível modificar a forma de um objeto / interface (então fazendo alterações e usando a linguagem) assim
 4. Todos *get* e *set* são métodos indiretos de um projeto ruim
 5. Interfaces (muitas vezes) não são indiretos de projeto ruim

Princípios de Projeto OO

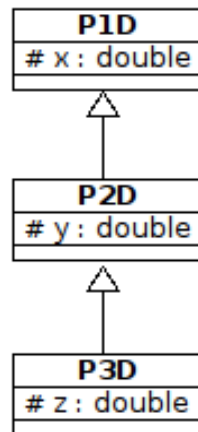
1. Prefira o método sobre a classe
2. Projete pensando em interfaces
3. Prefira a relação a herança
4. Princípio do Aberto-Fechado = *Open-Closed Principle*?
5. Princípio da substituição de Liskov = *Liskov Substitution Principle*?
6. Peça por ajuda, não por informação
7. *One Rule, One Place*

Princípios de Projeto OO

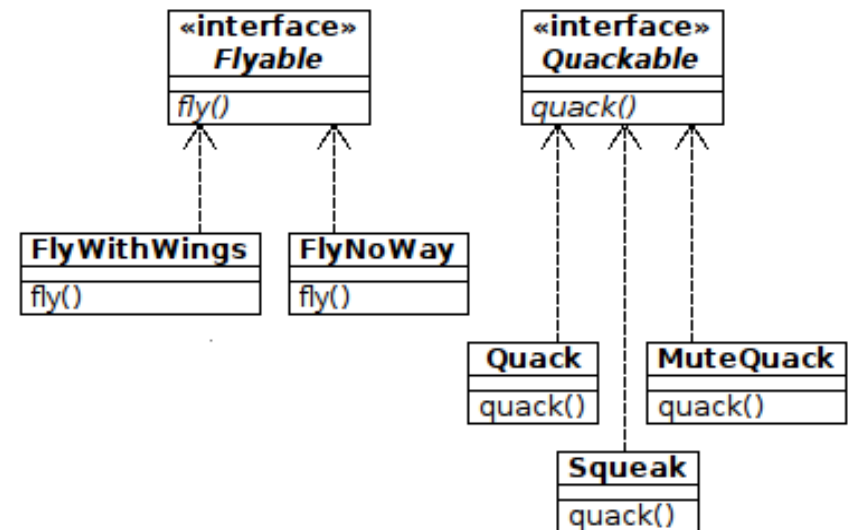
1

ENCONTRE O QUE VARIA E O ENCAPSULE

HERANÇA DE IMPLEMENTAÇÃO (sub-classing)



HERANÇA DE INTERFACE (sub-typing)



Princípios de Projeto OO

1

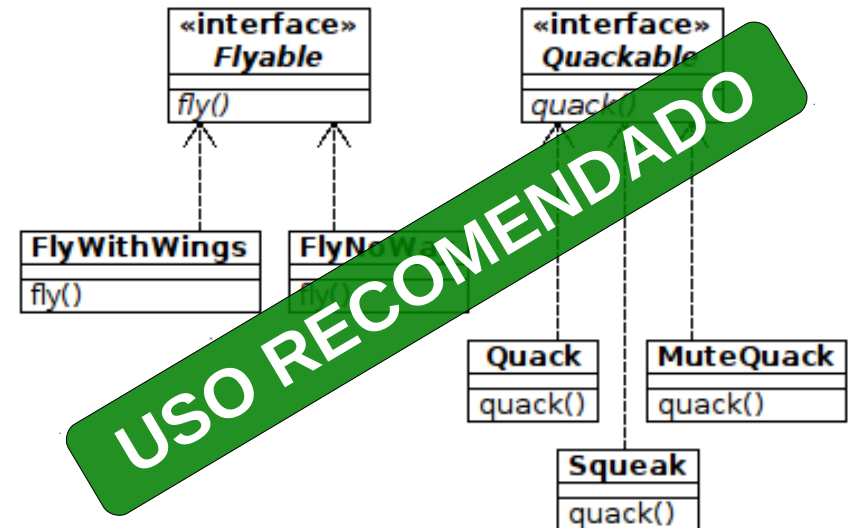
ENCONTRE O QUE VARIA E O ENCAPSULE

HERANÇA DE IMPLEMENTAÇÃO
(sub-classing)



USE COM CUIDADO

HERANÇA DE INTERFACE
(sub-typing)



USO RECOMENDADO

Princípios de Projeto OO

1

ENCONTRE O QUE VARIA E

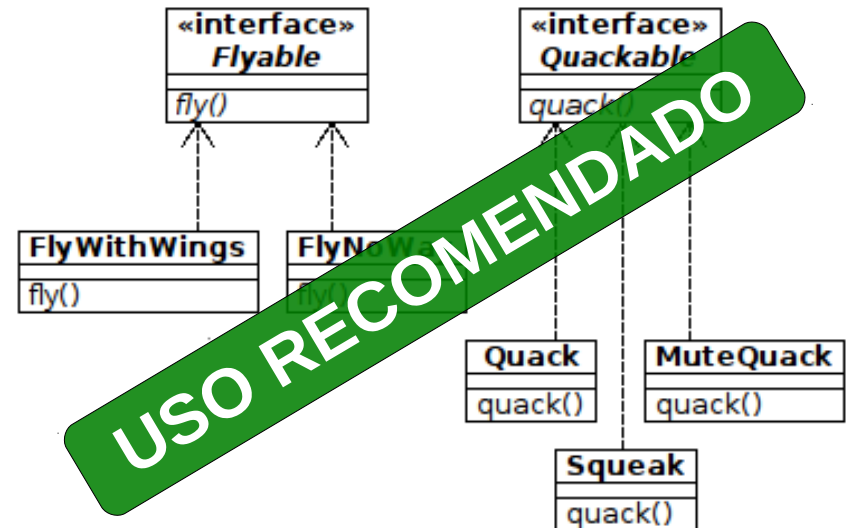
Quero saber mais !

HERANÇA DE IMPLEMENTAÇÃO
(sub-classing)



USE COM CUIDADO

HERANÇA DE INTERFACE
(sub-typing)

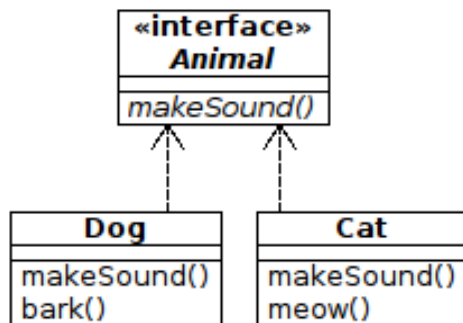


USO RECOMENDADO

Princípios de Projeto OO

2

PROGRAME PENSANDO EM INTERFACES



PROGRAMAÇÃO DIRIGIDA A IMPLEMENTAÇÕES

```
Dog dog = new Dog();
dog.bark();
```

PROGRAMAÇÃO DIRIGIDA A INTERFACES

```
Animal animal = new Dog();
animal.makeSound();
```

OU (AINDA MELHOR)

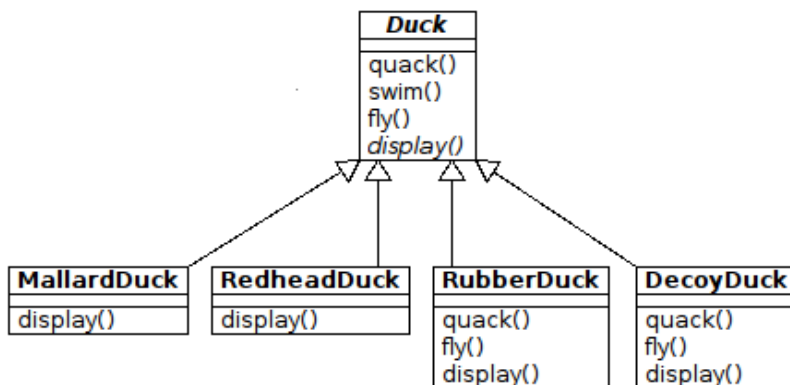
```
Animal animal = createAnimal();
animal.makeSound();
```

Princípios de Projeto OO

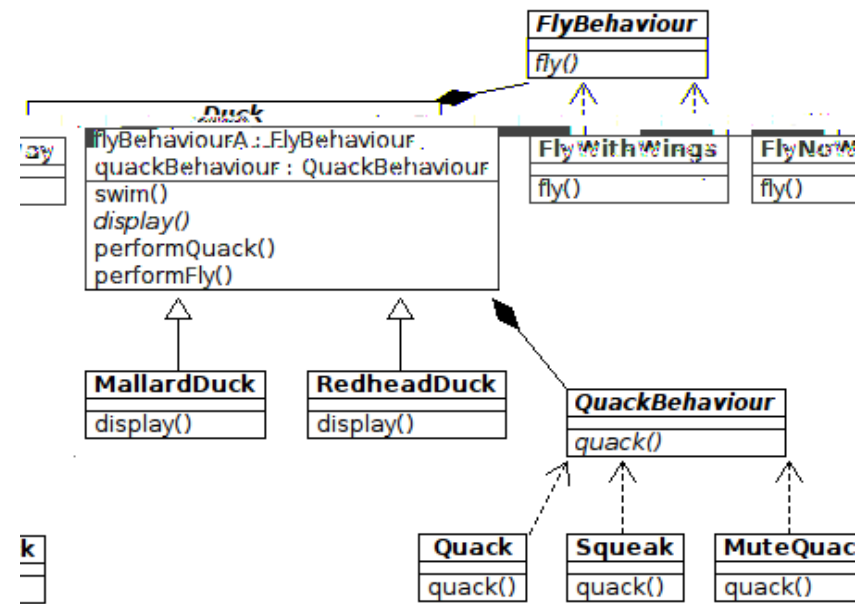
3

PREFIRA AGREGAÇÃO A HERANÇA

COM HERANÇA



COM AGREGAÇÃO

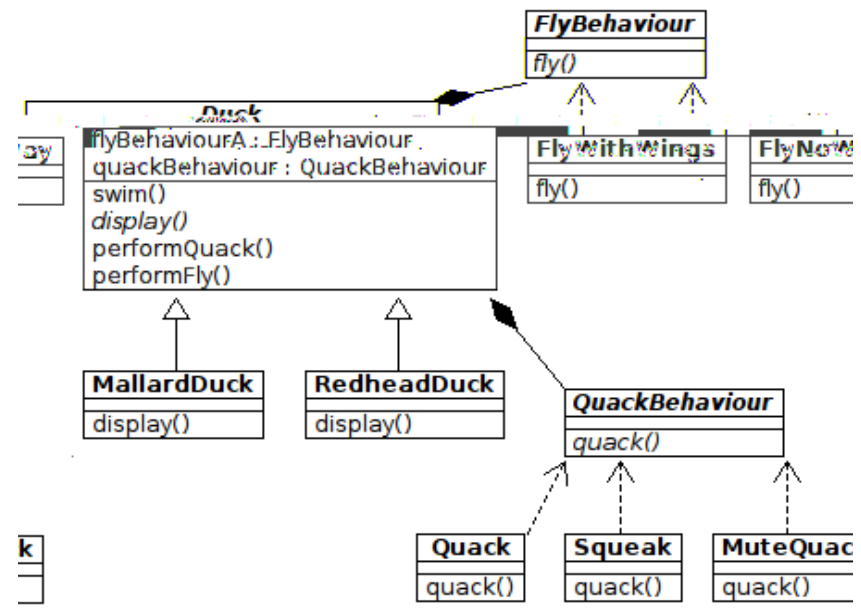


Princípios de Projeto OO

4

PRINCÍPIO DO ABERTO-FECHADO: módulos (classes) devem ser fechados para modificação e abertos para extensão

- No! as fun"ion(ões)idades são "riadas "o (a introdução de no! as ")asses
- %!ita a introdução de no! os *bugs*
- Funda (enta u (a ar ' uitetura de *plugins*



Princípios de Projeto OO

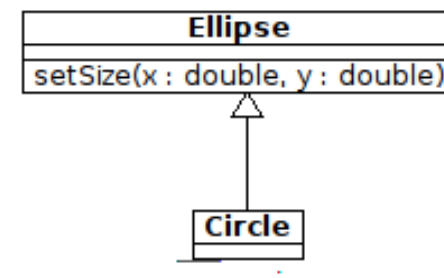
5

PRINCÍPIO DA SUBSTITUIÇÃO DE BÁRBARA LISKOV

Seja uma propriedade definida para objetos do tipo .
Então deve ser verdade para objetos do tipo , onde
é um sub-tipo de

- Defina o "on"eito de *substitutability*
- For (a)i5a a (etodo)o1ia de *design by contract*

VIOLAÇÕES DO PRINCÍPIO



Princípios de Projeto OO

5

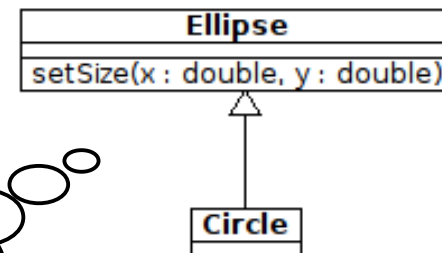
PRINCÍPIO DA SUBSTITUIÇÃO DE BÁRBARA LISKOV

Seja uma propriedade definida para objetos do tipo .
Então deve ser verdade para objetos do tipo , onde
é um sub-tipo de

Quero saber mais !

- Benfeito de *substitutability*.
- For (a) i5a a (etodo) o1ia de *design by contract*.

VIOLAÇÕES DO PRINCÍPIO



Quero saber mais !

Princípios de Projeto OO

6

PEÇA POR AJUDA, NÃO POR INFORMAÇÃO

- 8A (anutenibi)idade / in!ersa (ente propor"iona) < 'uantidadade de dados 'ue trafe1a entre os objetos9 [James Gosling]
- % (pou"os "asos *gets* e *sets* são justifi"ados

COM GETS E SETS

```
Money a, b;  
a.setValue(a.getValue() +  
           b.getValue());
```

SEM GETS E SETS

```
Money a, b;  
a.increaseBy(b);
```

Princípios de Projeto OO

6

PEÇA POR AJUDA, NÃO POR INFORMAÇÃO

- 8A (anutenibi)idade / in!ersa (ente propor"iona) < 'uantidadade de dados 'ue trafe1a entre os objetos9 [James Quero saber mais !
- % (pou"os e sets são just...

COM GETS E SETS

```
Money a, b;  
a.setValue(a.getValue() +  
           b.getValue());
```

SEM GETS E SETS

```
Money a, b;  
a.increaseBy(b);
```

Princípios de Projeto OO

7

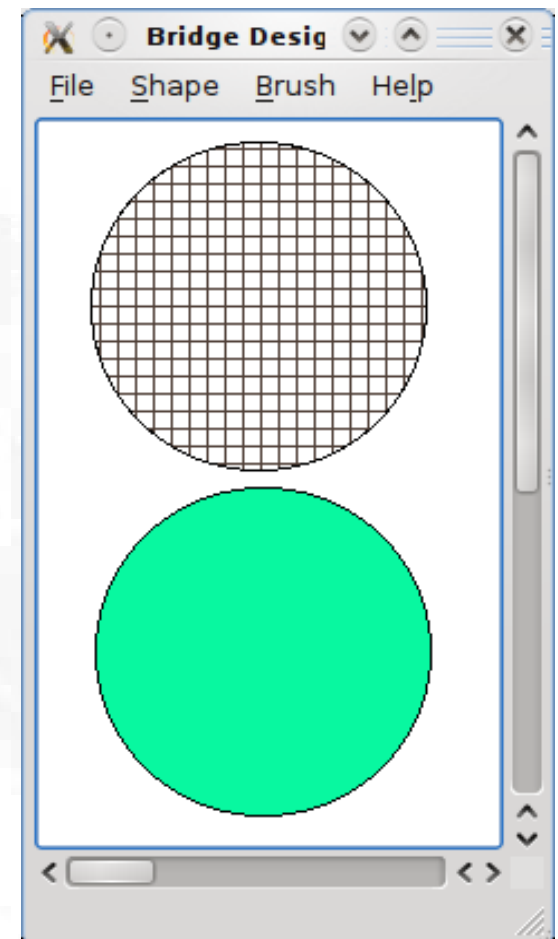
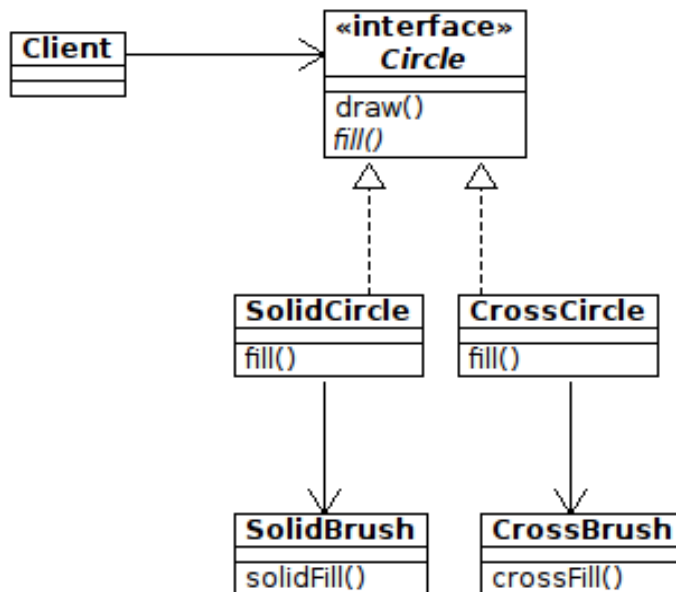
ONE RULE, ONE PLACE

- Cada regra de projeto, "deve" ser indicada (entrada e saída) em um único lugar, evitando redundância

Estudo de Caso:

- Contexto: O sistema de desenho possui dois tipos diferentes de preenchedores, sendo um usado

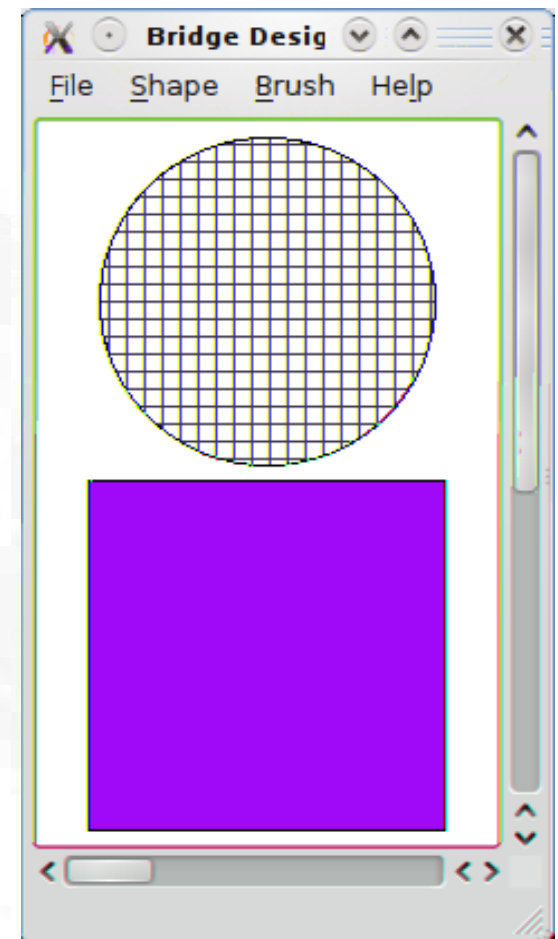
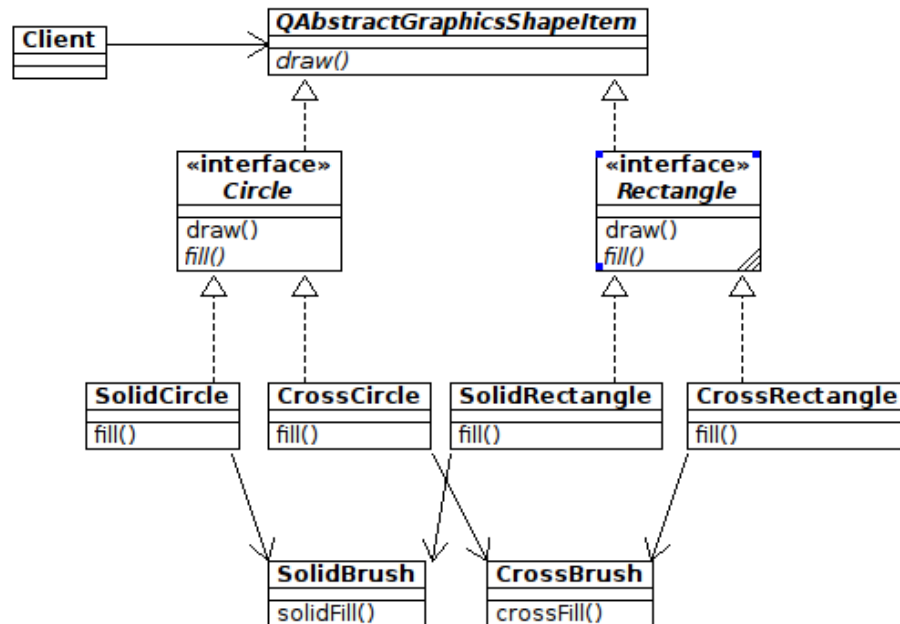
1ª SOLUÇÃO



Estudo de Caso:

- 1ª (udança o siste (a de! e a1ora ta (b/ (desen2ar retDn1u)os "o (as duas possibi)idades de preen"2i (ento

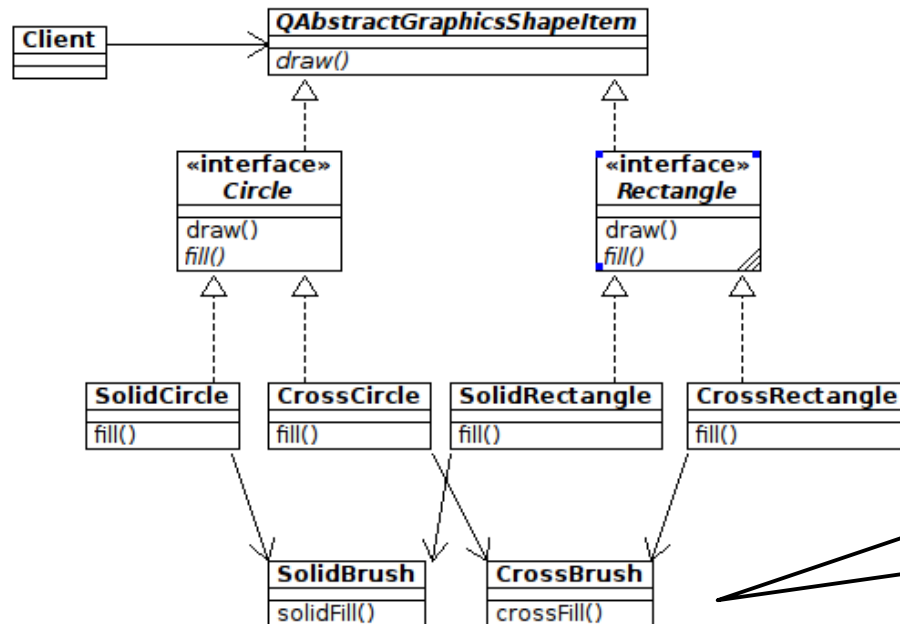
2ª SOLUÇÃO



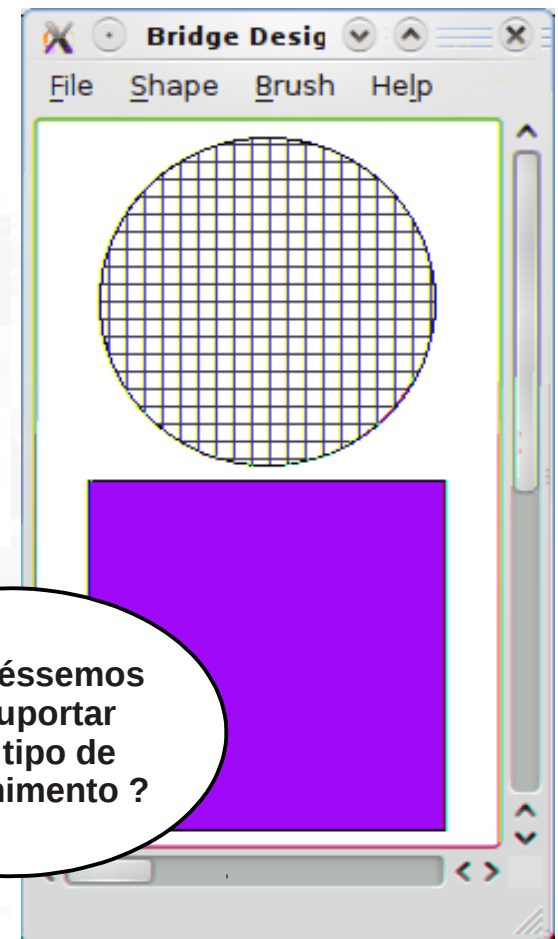
Estudo de Caso:

- 1ª (udança o siste (a de! e a1ora ta (b/ (desen2ar retDn1u)os "o (as duas possibi)idades de preen"2i (ento

2ª SOLUÇÃO

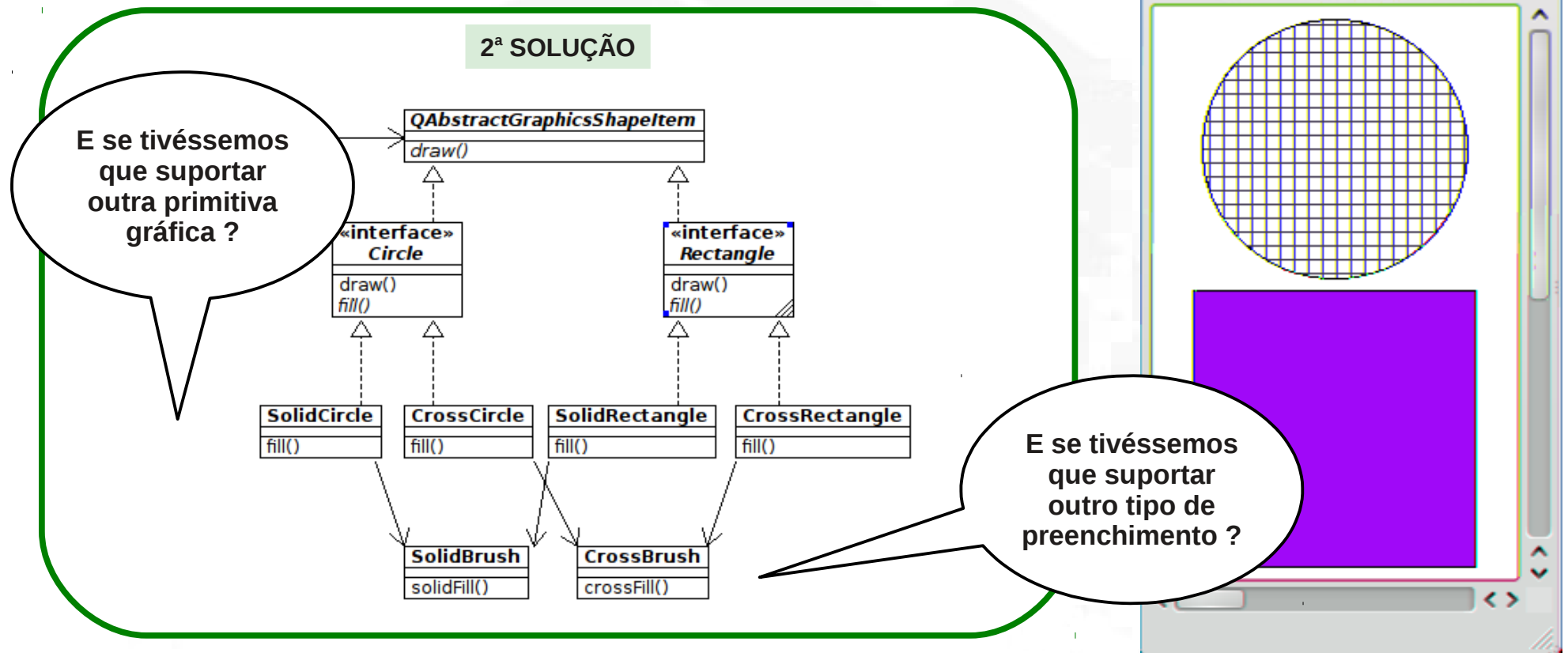


E se tivéssemos
que suportar
outro tipo de
preenchimento ?



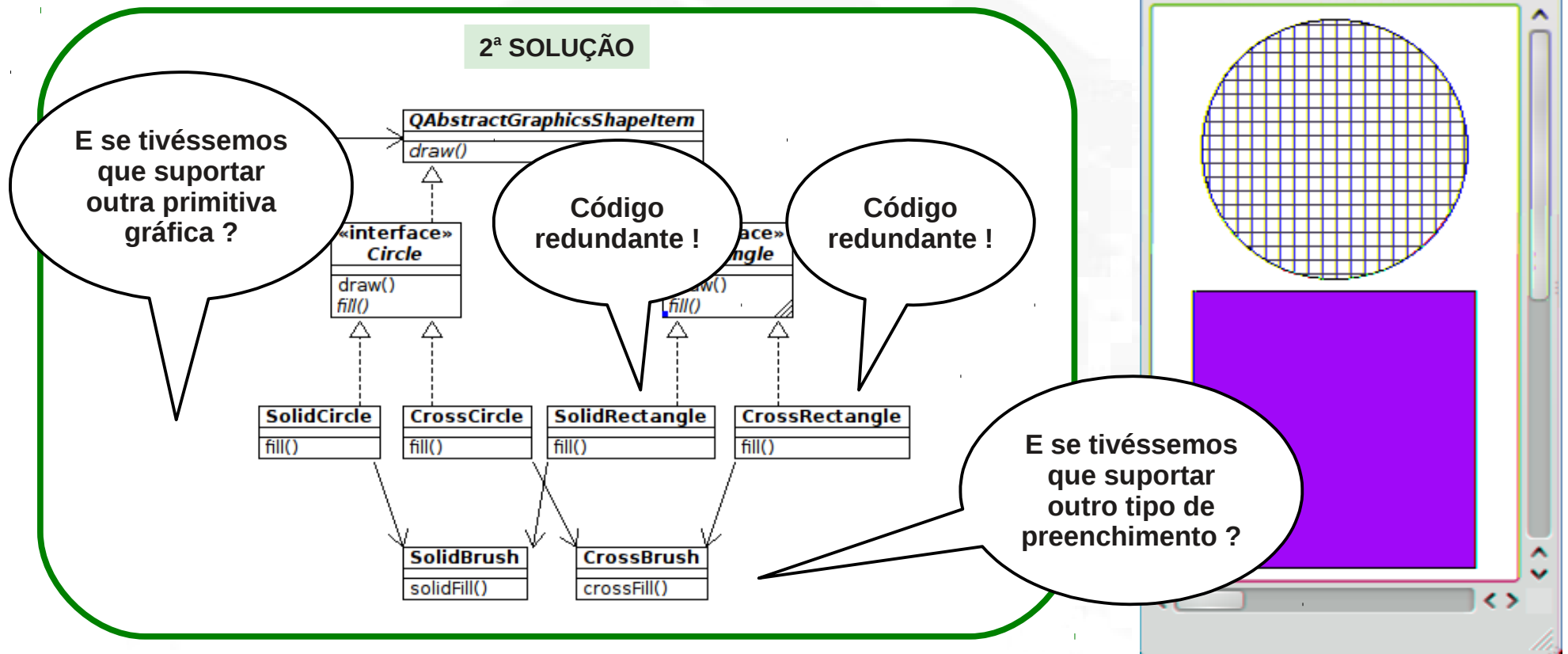
Estudo de Caso:

- 1ª (udança& o siste (a de!e a1ora ta (b/ (desen2ar retDn1u)os "o (as duas possibi)idades de preen"2i (ento



Estudo de Caso:

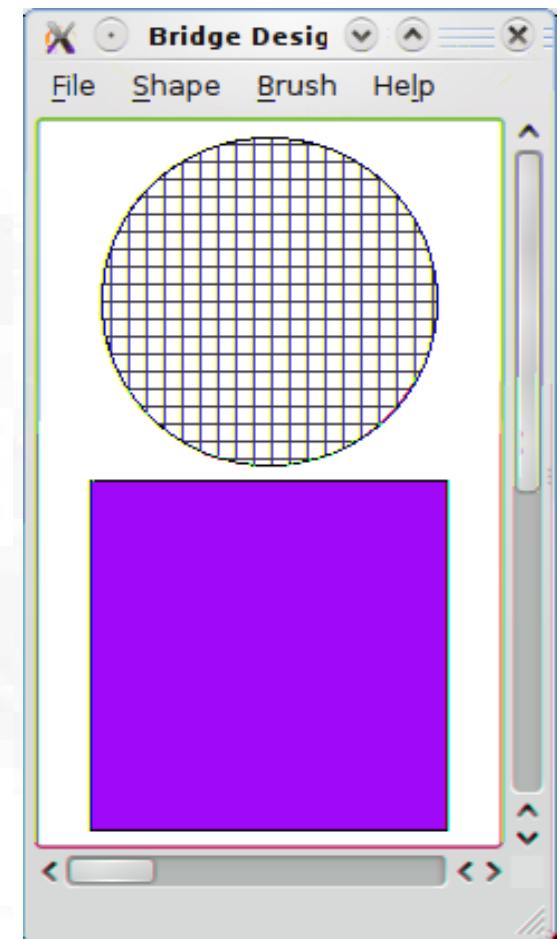
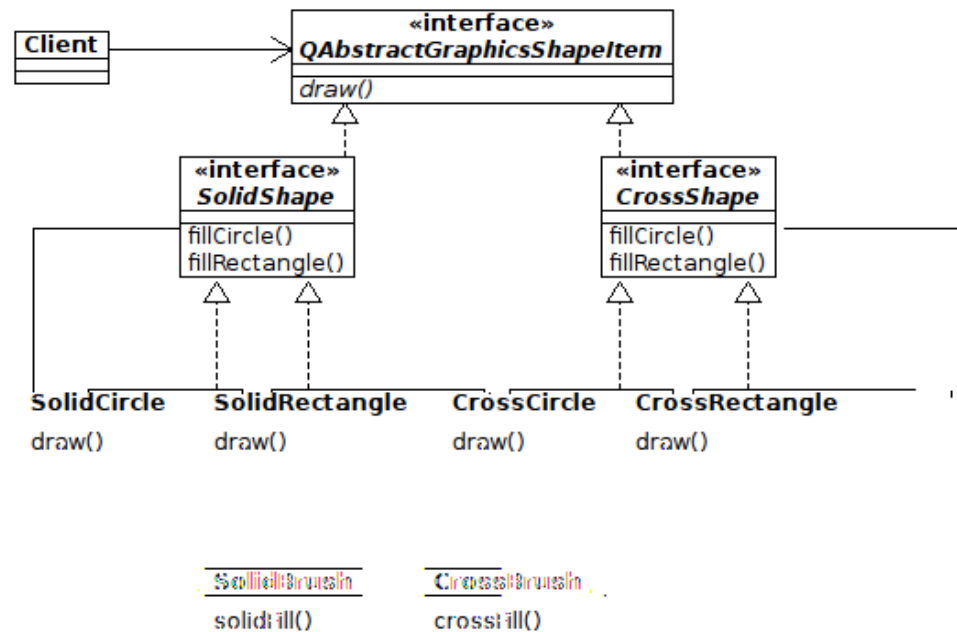
- 1ª (udança& o siste (a de!e a1ora ta (b/ (desen2ar retDn1u)os "o (as duas possibi)idades de preen"2i (ento



Estudo de Caso:

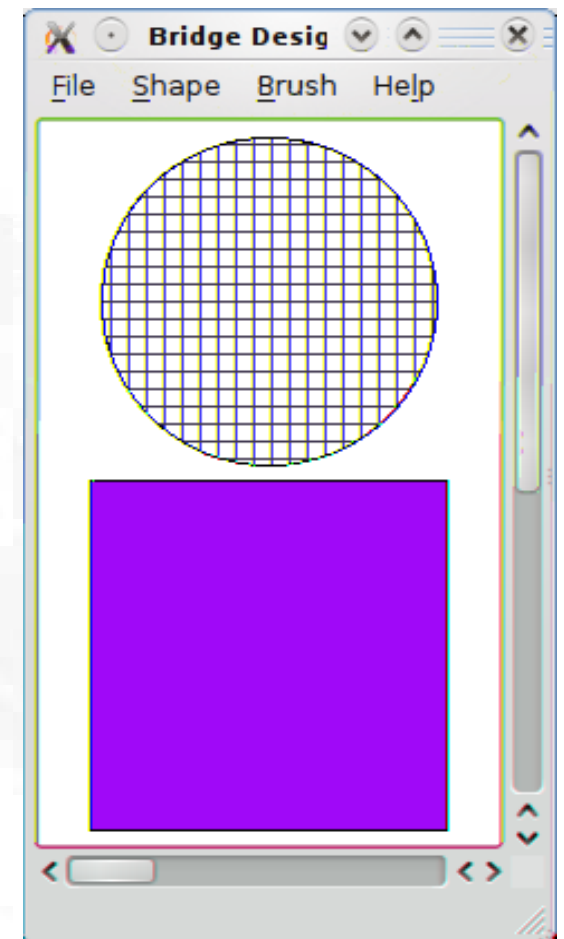
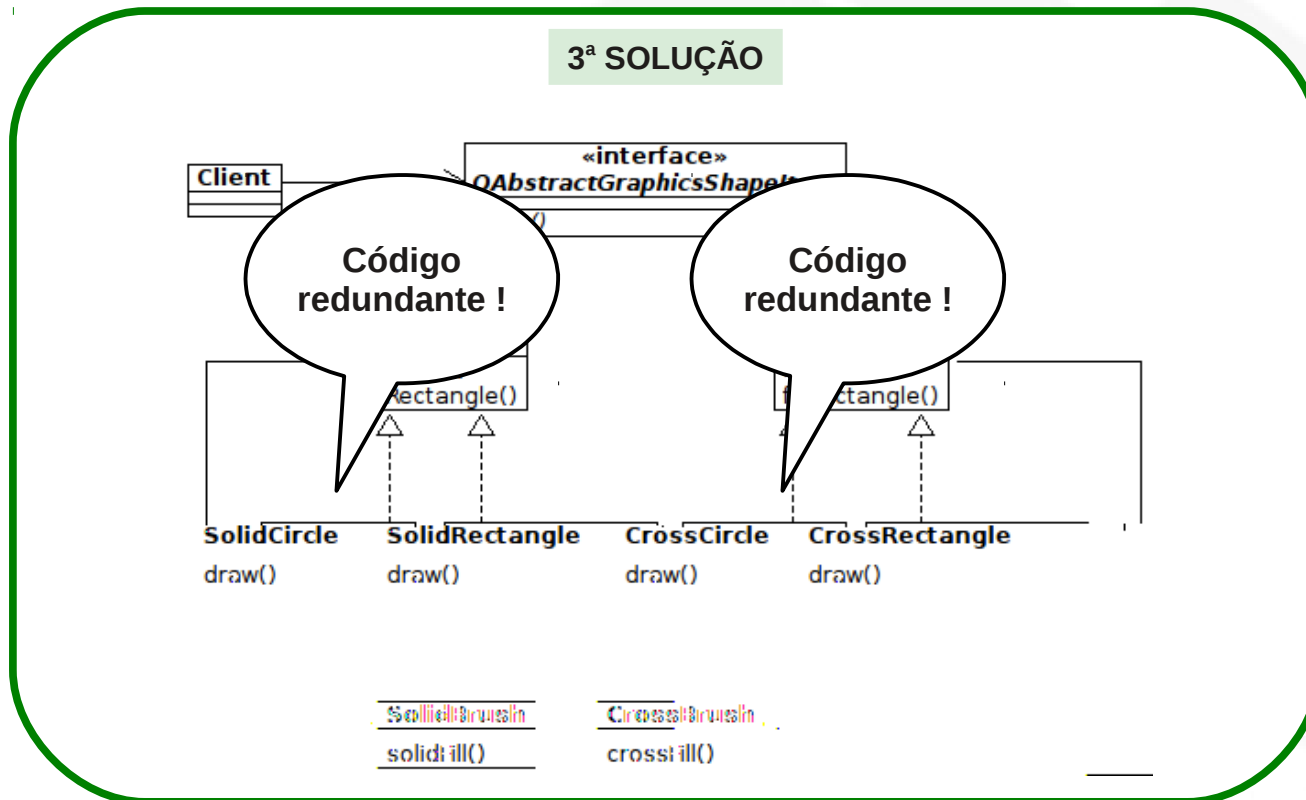
- 1ª (udança o siste (a de! e a1ora ta (b/ (desen2ar retDn1u)os "o (as duas possibi)idades de preen"2i (ento

3ª SOLUÇÃO



Estudo de Caso:

- 1ª (udança o siste (a de! e a1ora ta (b/ (desen2ar retDn1u)os "o (as duas possibi)idades de preen"2i (ento

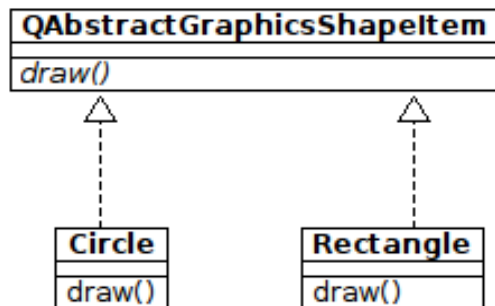


Estudo de Caso:

- \$ 'ue 2+ de errado "o (estes projetos :
 - - io)a (o prin"#pio ① 8%n"ontre o 'ue !aria e o en"apsu)e9 e o prin"#pio ③ 8Prefira a1re1ação a 2erança9
 - Luais aspe"tos !aria (:
 - As pri (iti!as 1r+fi"as& "#r"u)o0 retDn1u)o et"
 - As for (as de preen"2i (ento& s,)ido0 "ru5ado et"
 - Be!e-se "on"eber tais aspe"tos "o (o "on"eitos 'ue pode (possuir diferentes i (p)e (entações
 - I eranças de i (p)e (entação a"op)a (eterna (ente a ")asse-fi)2a "o (a ")asse-pai

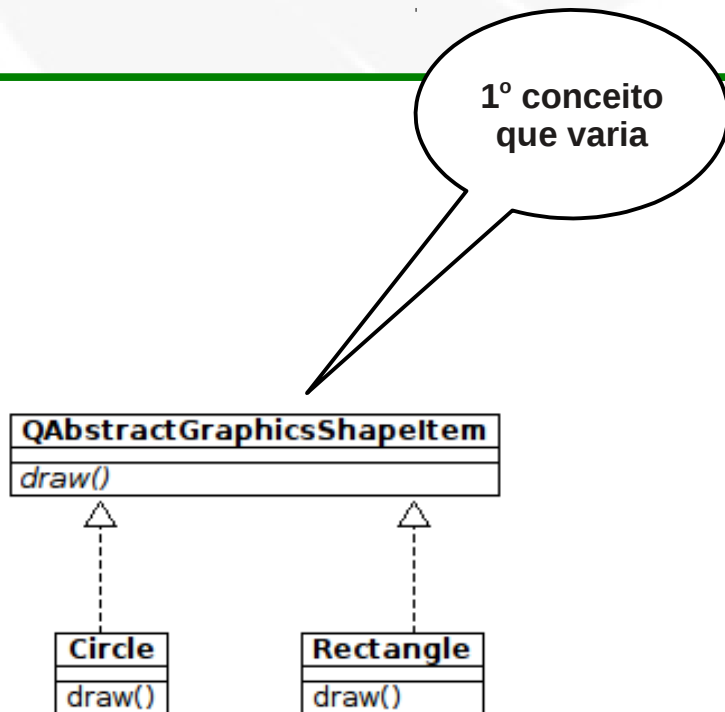
Estudo de Caso:

- * (projeto (e)2orado&



Estudo de Caso:

- * (projeto (e)2orado&



Estudo de Caso:

- * (projeto (e)2orado&



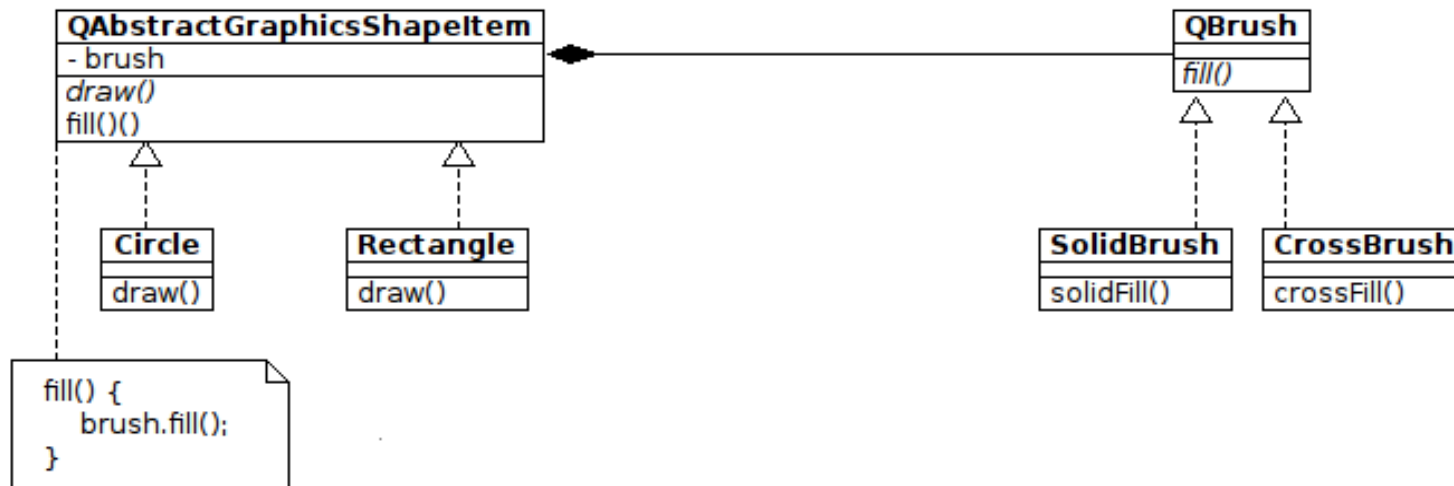
Estudo de Caso:

- * (projeto (e)2orado&



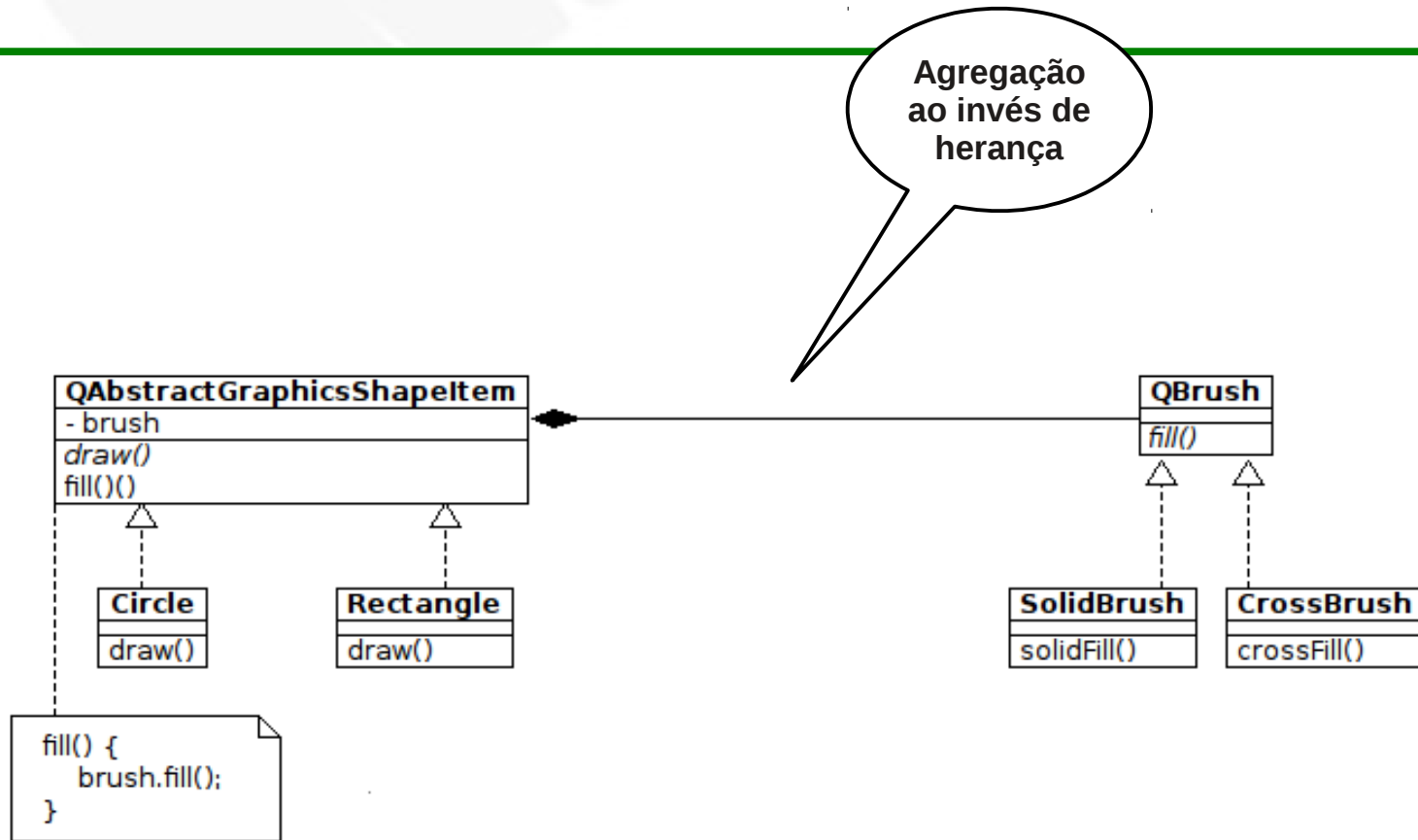
Estudo de Caso:

- * (projeto (e)2orado&



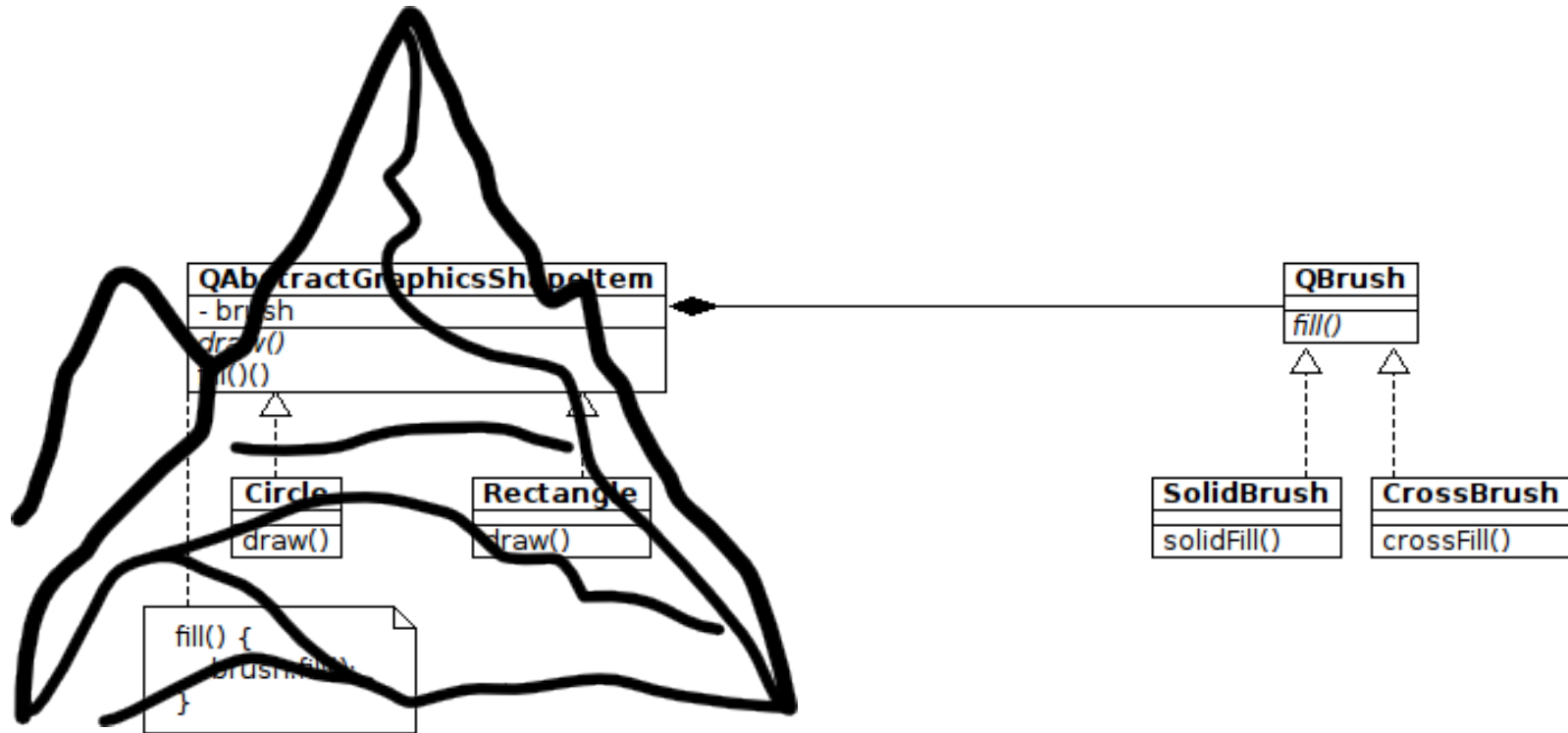
Estudo de Caso:

- * (projeto (e)2orado&



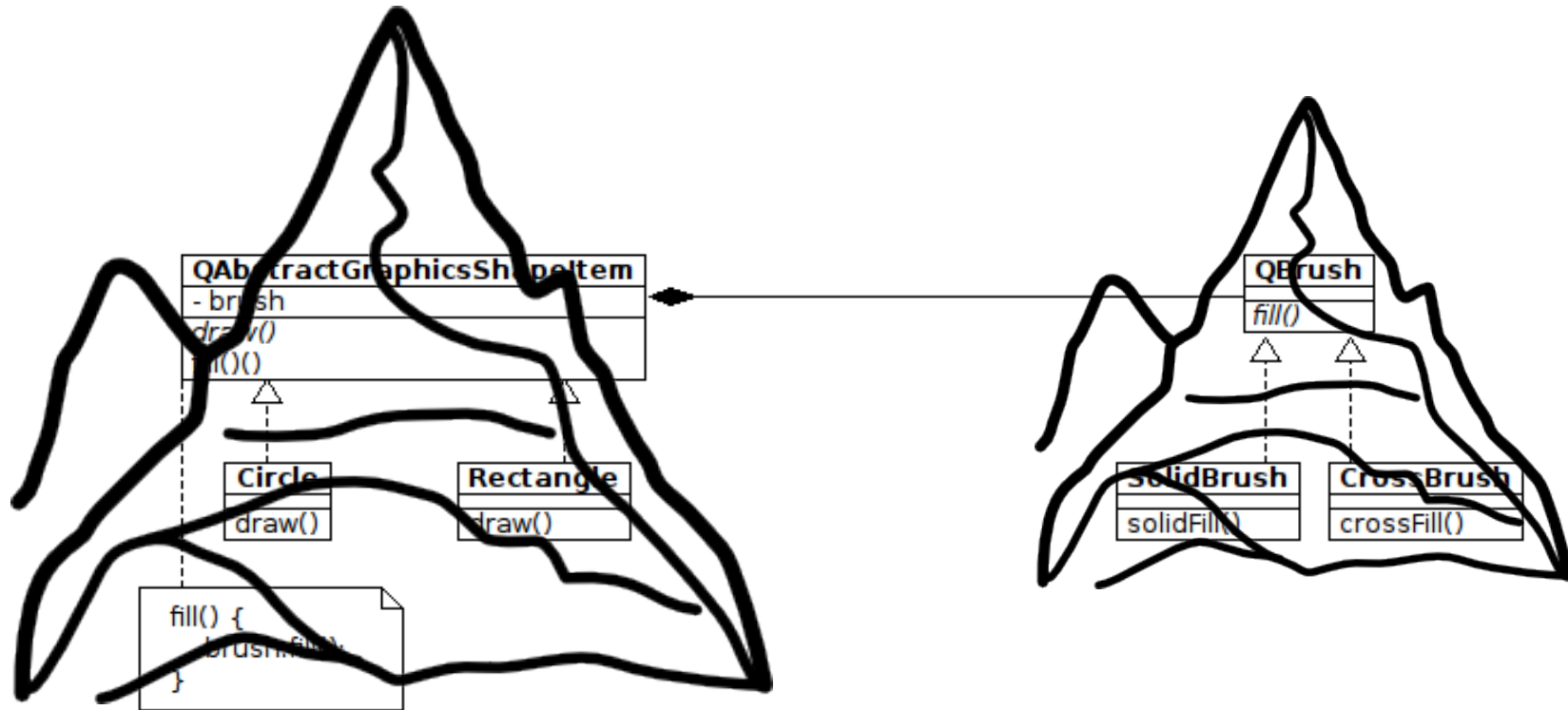
Estudo de Caso:

- * (projeto (e)2orado&



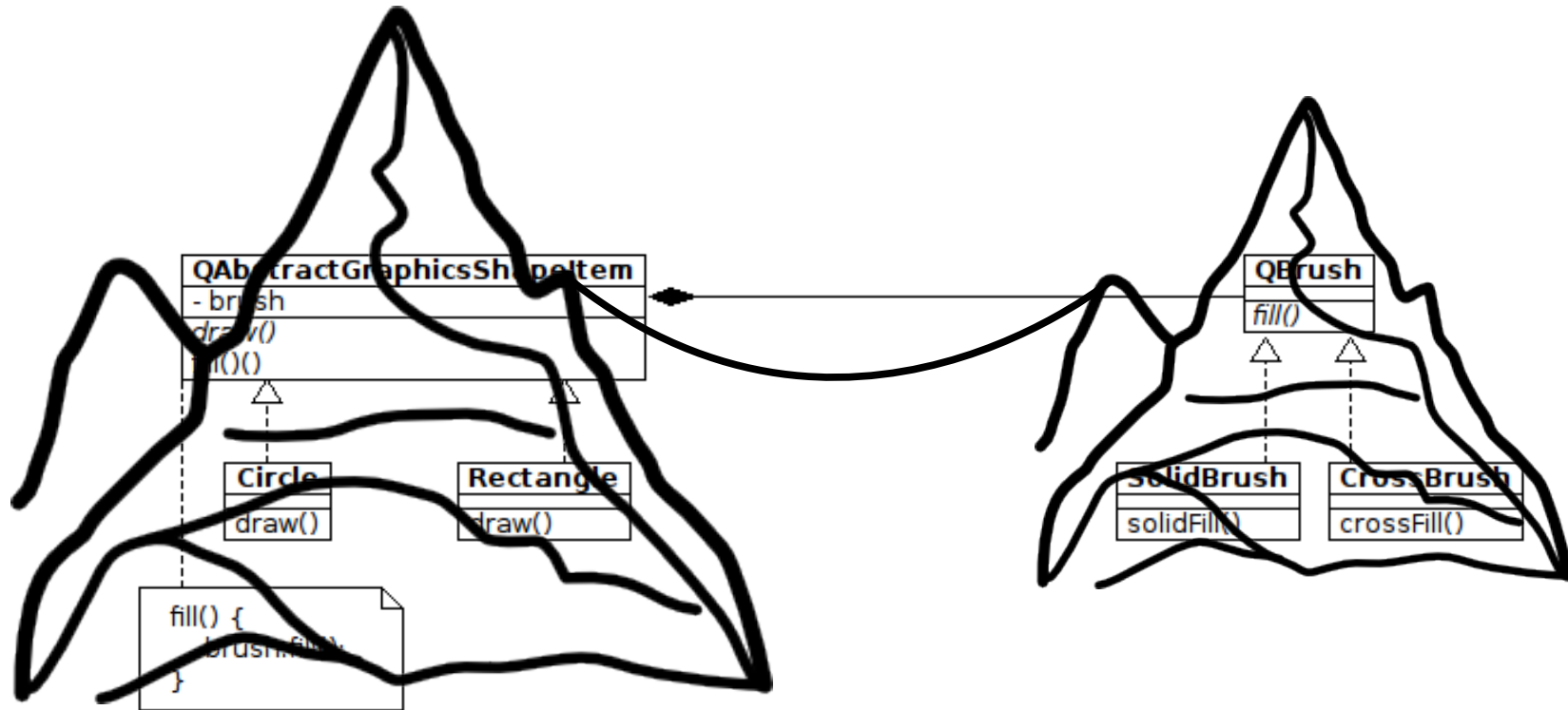
Estudo de Caso:

- * (projeto (e)2orado&



Estudo de Caso:

- * (projeto (e)2orado&



Estudo de Caso:

BRIDGE

Desacopla a abstração da sua implementação, de modo que os dois possam variar de forma independente

Padrões Arquiteturais e Idiomas

- Muitos outros padrões existem (e são aplicáveis em muitos contextos)
 - *Creational Patterns* descrevem como criar objetos, geralmente criando instâncias de classes do sistema, geralmente criando objetos. %x& *Abstract Factory*, *Builder* e *Prototype*
 - *Structural Patterns* descrevem como organizar estruturas (classes, atributos, métodos, etc.) de uma aplicação, geralmente descrevendo a organização de objetos, geralmente descrevendo a organização de objetos. %x& *Adapter*, *Bridge*, *Decorator* e *Flyweight*
 - *Behavioral Patterns* descrevem como organizar padrões de comportamento entre objetos, geralmente descrevendo a organização de objetos, geralmente descrevendo a organização de objetos. %x& *Strategy*, *Observer* e *Iterator*

Padrões Arquiteturais e Idiomas

- Padrões são aplicados e (diversos) meios de abstração

PADRÕES (ESTILOS) ARQUITETURAIS

PADRÕES DE PROJETO

IDIOMAS DE PROGRAMAÇÃO

Padrões Arquiteturais e Idiomas

- Padrões são aplicados e (diversos) meios de abstração

PADRÕES (ESTILOS) ARQUITETURAIS

PADRÕES DE PROJETO

IDIOMAS DE PROGRAMAÇÃO

Boas práticas para desenvolvimento em uma determinada linguagem

Padrões Arquiteturais e Idiomas

- Padrões são aplicados e (diversos) meios de abstração

PADRÕES (ESTILOS) ARQUITETURAIS

PADRÕES DE PROJETO

IDIOMAS DE PROGRAMAÇÃO

Soluções flexíveis
para problemas de
projeto OO
recorrentes

Boas práticas para
desenvolvimento em
uma determinada
linguagem

Padrões Arquiteturais e Idiomas

- Padrões são aplicados e (diversos) níveis de abstração

PADRÕES (ESTILOS) ARQUITETURAIS

Soluções recorrentes
para modularização
em alto nível

PADRÕES DE PROJETO

IDIOMAS DE PROGRAMAÇÃO

Soluções flexíveis
para problemas de
projeto OO
recorrentes

Boas práticas para
desenvolvimento em
uma determinada
linguagem

Conclusões

- Estudar padrões de projeto / utilizar a existente oportunidade para efetivamente entender os princípios fundamentais da orientação a objetos
- Existe uma solução de compromisso entre o uso de padrões de projeto e a possibilidade de *paralysis by analysis*?
- Categorias de padrões para diferentes tipos de problemas estão disponíveis = 300 reativos e sistemas distribuídos etc?
- Algumas bibliotecas tais como o L7 ou as APIs do Java fazem uso intensivo de padrões e devem ser estudadas
- * (padrão de projeto 1era) (então aparece e (conjunto "o (ou "o (a parte de? outro padrão

INF011 – Padrões de Projeto

00 - Apresentação

Sandro Santos Andrade
sandroandrade@ifba.edu.br

Instituto Federal de Educação, Ciência e Tecnologia da Bahia
Departamento de Tecnologia Eletro-Eletrônica
Graduação Tecnológica em Análise e Desenvolvimento de Sistemas

