

Aluno: _____

1ª Avaliação Individual - 2013.1

Questão 1) (2,0) Indique, no código-fonte apresentado abaixo, quais dos sete princípios de projeto OO vistos em sala foram violados e o local no código-fonte onde a violação ocorre. Utilize os números das linhas para indicar onde a violação ocorre. Indique porque o princípio foi violado e que tipo de problema essa violação pode trazer para a aplicação.

<pre> 01. abstract class Duck { 02. public abstract void fly(); 03. public abstract void quack(); 04. public Color getColor() { return color; } 05. private Color color; 06. } 07. class FlyableDuck extends Duck { 08. public void quack() { // no quack } 09. public void fly() { // fly implementation } 10. } 11. class QuackableDuck extends Duck { 12. public void fly() { // no fly } 13. public void quack() { // quack implementation } 14. } 15. class FlyableQuackableDuck extends Duck { 16. public void quack() { // quack implementation } 17. public void fly() { // fly implementation } 18. } </pre>	<pre> 19. class Main { 20. void handleQuack(FlyableDuck d) { 21. d.quack(); 22. } 23. public static void main (String argv[]) { 24. FlyableDuck d = new FlyableDuck(); 25. handleQuack(d); 26. } 27. void handleFly(Duck d) { 28. if (d instanceof FlyableDuck) { 29. ((FlyableDuck) d).fly(); 30. } 31. else if (d instanceof FlyableQuackableDuck) { 32. ((FlyableQuackableDuck) d).flyquacking(); 33. } 34. } 35. } </pre>
---	--

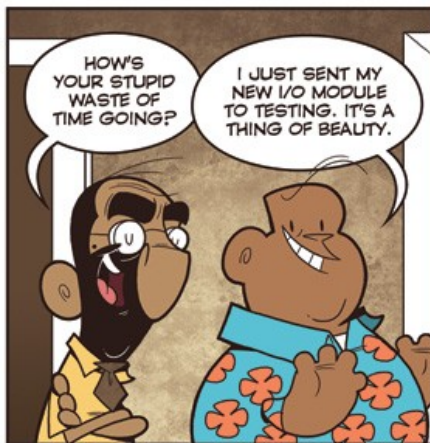
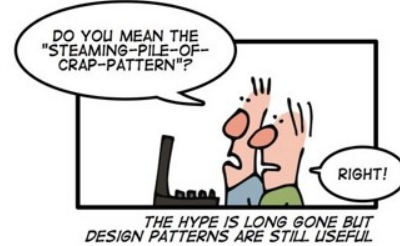
Questão 2) (2,0) Você trabalha em uma empresa que desenvolve um jogo para três diferentes *devices* (computadores *desktop*, *tablets* e *smartphones*). Um *game engine* diferente é utilizado para cada *device*, de modo a adequar a aplicação aos recursos disponibilizados pela plataforma. De forma similar, diferentes arquivos de áudio para trilha sonora e diferentes *backends* de entrada de dados (ex: *joystick*, *touch-screen* ou teclado) são utilizados, a depender do *device* sendo utilizado no momento. O código atual da inicialização do jogo é apresentado abaixo. Faça uma refatoração deste código, utilizando um *design pattern* que possibilite o suporte a novos *devices* requerendo a modificação de apenas uma única linha de código. Apresente o diagrama de classes e as assinaturas dos métodos que compõem a sua solução. Indique qual *pattern* foi utilizado.

<pre> class Game { public void init() { if (Settings.device() == Settings.Desktop) { DesktopEngine e = new DesktopEngine(); DesktopSoundTrack s = new DesktopSoundTrack(); DesktopInputBackend i = new DesktopInputBackend(); start(e, s, i); } else if (Settings.device() == Settings.Tablet) { TabletEngine e = new TabletEngine(); TabletSoundTrack s = new TabletSoundTrack(); TabletInputBackend i = new TabletInputBackend(); start(e, s, i); } } } </pre>	<pre> else if (Settings.device() == Settings.Smartphone) { SmartphoneEngine e = new SmartphoneEngine(); SmartphoneSoundTrack s = new SmartphoneSoundTrack(); SmartphoneBackend i = new SmartphoneBackend(); start(e, s, i); } } } </pre>
--	--

Questão 3) (2,0) Apresente um exemplo inédito de *Application Framework* que utilize o padrão *Factory Method*. Qual o papel do *Factory Method* na implementação da Inversão de Controle ? Apresente o diagrama de classes da sua solução e o código-fonte da classe da aplicação que implementa o *Factory Method*.

Questão 4) (2,0) Indique como a utilização de *Prototype Managers* podem flexibilizar ainda mais a solução que você apresentou na questão 2. Que tipo de flexibilidade adicional o jogo passou a apresentar após a utilização do *Prototype* ?

Questão 5) (2,0) Indique, nas três soluções apresentadas acima, quais princípios de projeto OO foram utilizados e o local onde eles aparecem.



Not Invented Here™ © Bill Barnes & Paul Southworth

NotInventedHere.com

Boa Sorte !