PILARES DA POO: ENCAPSULAMENTO	PILARE
O que são os Pilares da POO ?	Pilares da POO: São os toda a sua infraestrutu funcionaria, ou ainda par Encapsulamento; 2 Herança; Esses são os 3 pilares os "Abstração" como tar Abstração é um conce dos programadores nã
O que é o conceito de Encapsulmento ?	Encapsulamento: é código tanto da infl encapsulada, quant o código externo. P através de uma Pilh elétrica transitar so perigoso o contato encapsulada, para o danos ao mundo ex
- reservada para a questão acima -	pasta MnO² da pilhi protegido de código o desenvolvedor na código é trancafiad código externo a menos que seja referenciado. Além disso, assim como u
- reservada para a questão acima -	pilha pode ser utiliz nos mais variados equipamentos elétr os códigos encapsu podem ser referen e reutilizados nos n variados tipos de aplicações.
Quais são alguns conceitos importantes do Encapsulamento em POO?	Dentro do Encapsulament - Mensagem: é como char - Visibilidade: podem exis Pública (que pode ser vist Encapsulamento); - Interface: a lista de servi serviços o encapsulament - Abstração: é o produto o resultados diferentes qua esses resultados são as ab
Quais as vantagens do Encapsulamento?	- Tornar mudanças inv sem que o usuário fina realidade estão aconto - Facilitar a reutilizaçã vários códigos diferen - Reduzir efeitos colar destruídos ou removic além disso, podemos que códigos fora do pa
O que são os Métodos Abstratos ?	Métodos Abstratos encapsulamento, o instâncias para que "abstratos" por que por através das inst transformadas de a Geralmente os mét
Como fazemos para encapsular os nossos códigos?	Podemos encapsular o "classes" que os "méto - Classes: equivale a nu encapsulamos nossos somos obrigados a rere - Métodos: equivale a nossas operações que inclusive podemos usa métodos muito mais e
Métodos getters e setters podem ser vistos como métodos abstratos?	Não! devemos lemb que estão presente acessados pelos usu "Métodos Especiais possamos acessar a métodos abstratos. setter não são méto

ES DA POO: ENCAPSULAMENTO

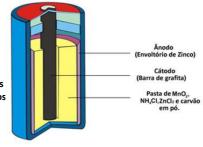
os conceitos fundamentais da POO que a sustentam e mantém tura de funcionamento. Sem esses pilares, a POO não pior, não existiria. Quais são esses pilares? Eles são: 3º Polimorfismo;

da POO - algumas publicações consideram o conceito de ambém sendo um dos pilares da POO, porém, como a eito existente dentro do "Encapsulamento", a grande maioria ñão a consideram um dos pilares.

é o conceito que visa proteger uma parte específica do fluência externa de um desenvolvedor sobre a parte ito da influência interna do código encapsulado sobre Podemos ilustrar a função do "Encapsulamento" por ha, o material principal da pilha que faz a corrente obre um equipamento é a pasta de MnO², mas seria externo com essa pasta, por isso a pilha é que ela possa desempenhar seu papel sem causar xterior e nem permitir que o exterior cause danos a na. O mesmo se dá com o código encapsulado, ele é go externo para que

não possa causar danos ao código, ao mesmo tempo o do dentro de uma cápsula para que não interfira no

uma zada tricos, ulados nciados



nto, vamos ouvir falar dos seguintes conceitos:

- amamos a comunicação do código encapsulado com o mundo exterior;
- istir 2 tipo de visibilidade dentro de um Encapsulamento, visibilidade sta de fora do Encapsulamento) e Privada (só pode ser vista dentro do
- viços oferecidos por um encapsulamento, onde, por através destes nto pode entrar em contato com o mundo exterior;
- de um encapsulamento, ou seja, as instâncias (ou objetos) podem gerar ando utilizam os métodos presentes nas interfaces do Encapsulamento, bstrações;
- nvisíveis: podemos causar alterações nos resultados do código, nal fique á par de onde acontecem as mudanças, que na tecendo dentro da cápsula;
- ão de código: podemos reutilisar o código encapsulado em ntes que tenham a mesma interface da cápsula;
- iterais sobre o código: evitamos que códigos essenciais sejam idos sem guerer por desatenção, pois ele estará encapsulado. padronizar a interface de entrada da nossa cápula, evitando padrão danifiquem nossa aplicação:

s: São os métodos que compõem a Interface de um ou seja, aqueles métodos que estarão disponíveis ás e elas possam ser utilizá-los. Eles recebem o nome de e o objetivo deles é justamente gerar uma "abstração" stâncias, por através deles as instâncias podem ser acordo com o método escolhido pelo usuário: todos abstratos são públicos.

os códigos por através de 2 ferramentas UML, que são as

- nossa Interface de Encapsulamento, por através dela nós métodos do mundo externo, onde, para acessar os métodos eferenciar á Classe;
- aos nossos métodos abstratos, que usamos para encapsular as e estarão disponíveis na Interface de Encapsulamento, sar os métodos getters e setters para tornar os nossos encapsulados;

ibrar que os métodos abstratos são aqueles métodos es na nossa Interface de Encapsulamento para serem suários finais, já os métodos getters e setters que são is" são usados por nós desenvolvedores para que aos atributos privados por através dos nossos s. Portanto podemos dizer que os métodos getters e todos abstratos, eles são "Métodos Especiais".