**Anotações de APS**

* **O que é projeto?**

Age no que é abstrato, tem ênfase NO QUE, e como esse OQ, pra q serve e como se relaciona

com os outros objetos no sistema, sabendo que se modificar pode dar problema no futuro. então

ele é uma preocupação com o futuro. quem projeta algo, é uma projeção, pensar a frente.

É pensar no como uma classe e os objetos dela, se relacionam com outra classe e os objetos dela.

pensar no conjunto. Se fosse pensar somente em 1 coisa, seria implementação. projeto são vistos

externamente.

* **O que é implementação?**

Age no que é concreto, ênfase em COMO será feito e se preocupa com o presente.

É tudo que não é visto externamente, envolve a "arrumação" do código.

* **Onde encontrar princípios de projetos?**
  + livro Agile Software Development: principles, patterns, and practices
  + c2.com
  + htts://www.cs.cmu.edu/afs/cs.cmu.edu/pro
* **SOLID**
  + Single responsibility principle (SRP) - princípio da responsabilidade única
  + Open/close Principle (OCP) - principio aberto/fechado
  + Liskov Substitution Principle (LSP) - princípio de substituição de liskov
  + Interface Segregation Principle (ISP) - princípio da segregação da interface
  + Dependency Inversion Principle (DIP) - princípio da inversão da dependência
* **Padrões tem 4 elementos:**

1. Nome: referência útil para o vocabulário compartilhado
2. Problema: situações e contexto aplicáveis
3. Solução: os elementos que compõe o padrão
4. Consequências: vantagens e desvantagens

* **Definições**
* Abstrato: o que é genérico, não específico. Ex.: animais, pois podem ser mamíferos, anfíbios, etc.
* Acoplamento: é o quanto uma classe sabe/depende das outras.
* Agregado: é um tipo de associação que precisa complementar um objeto, mas não obrigatoriamente. Ex.: cunhado não de sangue.
* Atributos: são os elementos que definem a estrutura de uma classe, ou seja, variáveis.
* Classe: é uma descrição que abstrai um conjunto de objetos com características similares.
* Composto: é o quando uma classe sabe/depende de outra, se uma deixa de existir, todas também deixarão, vivem e morrem juntas.
* Concreto: o que é específico, não genérico.
* Construtor: determina como será a criação do objeto através de um parâmetro(definição).

* Gráfico UML: Linguagem de Modelagem Unificada, usada para modelagem da estrutura do projeto.
* Herança: herda todas classes acima, pois quando se cria herança, cria uma arvore da classe pai para as filhas.
* Interface: define o comportamento esperado de um conjunto de classes.
* Métodos/Funções: definem o comportamento dos objetos de uma classe. Funções são invocadas adicionando “( )” no final. Ex.: jogar( )
* Objeto: é o q possui um valor, pode ser uma variável, função ou dados.
* Traversing: navegar no grafo de objetos