ESCOLA POLITÉCNICA

Disciplina: TECNOLOGIA DE CONSTRUÇÃO DE SOFTWARE (CMP 1069)

Prof. MSc. Anibal Santos Jukemura

**Implementação N1 – peso 6**

Segue contexto para implementação em JAVA:

**História:**

Optimus Prime, líder dos Autobots, na tentativa de escapar da Guerra travada em Cybertron contra os Decepticons liderados pelo implacável Comandante Megatron, teve sua nave denominada A ARCA avariada gravemente. Os esforços dos Autobots para recuperarem a nave foram insuficientes e tiveram que fazer uma aterrisagem forçada no planeta Terra.

Imagem digital fictícia de personagem de desenho animado

Descrição gerada automaticamente com confiança média

Imagem digital fictícia de personagem de desenho animado

Descrição gerada automaticamente com confiança média

Uma imagem contendo mesa, computador

Descrição gerada automaticamente

Junto com o Líder Autobot, vieram também duas classes de soldados combatentes:

* Os CARBOTS, representados pelos soldados Bumblebee e Ironhide.
* OS AERIALBOTS, representados pelos tenentes Silverbolt e Skydive.

Um Autobot específico, o Oficial Ratchet também sobreviveu ao acidente da ARCA devido às suas habilidades médicas (ou mecânicas...).

**Bumblebee:**

Desenho de carro azul

Descrição gerada automaticamente com confiança baixa

Bumblebee é um soldado de elite, tático, espião, especialista em combate corpo-a-corpo, braço direito de Optimus Prime, seu melhor amigo.

Suas características especiais são:

Calendário

Descrição gerada automaticamente com confiança baixa

* Ataque: 4
* Vida: 9
* Defesa: 2
* Especialidade: ataque corpo-a-corpo.

**Optimus Prime:**

Motor de carro azul

Descrição gerada automaticamente com confiança baixa

Líder Autobot. Carrega em seu peito a Matriz da Liderança, artefato capaz de recriar a vida cibernética de inúmeros cybertronianos. Suas características são:

Uma imagem contendo ônibus, televisão, display, telefone

Descrição gerada automaticamente

* Ataque: 3
* Vida: 10
* Defesa: 2
* Especialidade: estrategista em combates.

**Ironhide:**

Capitão Autobot, especialista em armas e combate terrestre.

Imagem digital fictícia de personagem de desenho animado

Descrição gerada automaticamente com confiança baixa

Suas características são:

Uma imagem contendo texto, carro, comida, mesa

Descrição gerada automaticamente

* Ataque: 5
* Vida: 12
* Defesa: 2
* Especialidade: armas pesadas.

**Silverbolt:**

Imagem digital fictícia de personagem de jogo de vídeo game

Descrição gerada automaticamente com confiança baixa

Aerialbot supersônico. Líder dos Aerialbots. Comandante estrategista.

Suas características são:

Uma imagem contendo Calendário

Descrição gerada automaticamente

* Ataque: 4
* Vida: 12
* Defesa: 2
* Especialidade: emboscadas aéreas.

**Skydive:**

Skydive é provavelmente o piloto Autobot mais habilidoso que existe. Seu conhecimento de guerra aérea e tática é incomparável.

Imagem digital fictícia de personagem de desenho animado

Descrição gerada automaticamente com confiança baixa

Suas características são:

Interface gráfica do usuário

Descrição gerada automaticamente

* Ataque: 3
* Vida: 8
* Defesa: 2
* Especialidade: táticas aéreas de guerra.

**Ratchet:**

Imagem digital fictícia de personagem de jogo de vídeo game

Descrição gerada automaticamente com confiança baixa

Ratchet é o Oficial Médico dos Autobots e provavelmente o mais bravo dentre todos eles.

Suas características são:

Uma imagem contendo caminhão, carro, ônibus, andando de

Descrição gerada automaticamente

* Ataque: 3
* Vida: 14
* Defesa: 0
* Especialidade: engenharia cybertroniana.

Esses são os AUTOBOTS!!!

**Logotipo

Descrição gerada automaticamente**

Descrição da atividade:

Diante do contexto apresentado, desenvolva uma aplicação em JAVA que utilize o Padrão Criacional Factory Method para MELHOR descrever os autobots. Para isso, siga as seguintes exigências:

* CONSIDERE que existem quatro classes(TIPOS) de Autobots:
  + um LIDER: Optimus
  + um MEDICO: Ratchet
  + dois AERIALBOTS: Sylverbolt e Skydive
  + dois CARBOTS: Bumblebee e Ironhide
* Defina a melhor metodologia a ser aplicada: Full Factory Method, Half Simple Method ou Simple Method.
* Descreva o diagrama UML de classes para representar a solução final implementada (isso vai ajudar muito!!!).
* Organize o código em três pacotes (packages): um pacote para o programa principal, um pacote para o modelo (usado para descrever os autobots) e outro pacote para representar a(s) Factory(ies). Exemplo:

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

* **Definição do MODELO:**
  + Para a definição do modelo Autobot, considere uma classe abstrata principal denominada Autobot, que possui um método concreto denominado **public void assembleAutobot()** e um **public abstract void printCardAutobot (String nome)**. O método assembleAutobot() deve somente imprimir uma mensagem: “Invocando Autobot...”.
  + Para as classes concretas, considere que cada uma irá gerar um autobot específico (seis autobots, no total). Cada uma dessa classes deve implementar o método abstrato da Classe Autobot denominado **public void printCardAutobot**(String nome). Esse método deve imprimir o nome do Autobot, TIPO, seguido da sua FICHA TÉCNICA (conforme apresentado na história);
* **Definição da(s) Factory(ies):**
  + Use classes abstratas nos locais apropriados, exceto para a definição da Factory principal que deve ser uma interface. Essa interface deve se chamar AutobotFactory e deve declarar um método denominado: **public Autobot summonAutobot(String robot)** . Além disso, essa fábrica deve declarar um método: **public abstract Autobot createAutobot(String robot).**
  + Cada factory utilizada deve implementar um método (concreto) do método **public abstract Autobot createAutobot(String robot)** definido pela AutobotFactory. Utilize-se da String robot para definir qual Autobot criar. Também deve implementar um método (concreto) do método **public Autobot summonAutobot(String robot).** Esse último método deve invocar os métodos **assembleAutobot()** e **createAutobot(String Robot)** e criar/retornar o autobot recém-criado.
* Para testar o programa, crie todas as fábricas necessárias, crie os autobots das fábricas respectivas e imprima as informações de todos os autobots. Segue o exemplo de uma das saídas:

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente