***Importante***: Para item abaixo deve ser copiado trechos do código que cumprem o requisito e explicado, se não for aparente, o porquê o requisito é cumprido. Sejam bem explícitos. Deve ser indicado também o arquivo da classe em que está o trecho do código. Eu avaliarei o código do Github a partir desse documento para confirmá-lo e também para detectar possíveis erros. **Quem não seguir o que está indicado aqui, não terá o projeto avaliado e perderá a atividade.**

**Requisitos de implementação**

***Todos os atributos e funções membros devem estar relacionados a classe***

1. Pelo menos 4 atributos (Todos as classes do pacote personagem cumprem o requisito)

**Classe Personagem**

public class Personagem {

int id; //

static String nome;

int saude;

Item[] Itens;

private static final int forcaMaxima=100;

1. Pelo menos 4 funções membros sem incluir get e set

**Classe Personagem**

public int Curar(int saude)

public int reduzirSaude(int saude)

public void print(){

public void checkSaude(int saude

**Requisitos de implementação**

1. Cinco classes: uma superclasse e duas subclasses, e duas classes relacionadas ao projeto

Super Classe : Personagem;

SubClasses: PersonagemMilitar, Legionario;

Classes Relacionadas : Item, Esquadrao.

1. Sempre usar o super para o máximo de reaproveitamento de código

**Classe PersonagemMilitar** (Construtor, Linha 19)

public PersonagemMilitar(int ataque, int defesa, int corrida,int id, String nome, int saude,Item[] Itens) {

super(id, nome, saude, Itens);

this.ataque = ataque;

this.defesa = defesa;

this.corrida = corrida;

}

1. Diagrama de classes (obrigatório salvar também o png do diagrama no gitHub)

Diagrama de Classes, juntamente com o arquivo imagem em png

Está no diretório /legionários\_program/Diagrama UML/

1. Todos os atributos devem ser inicializados. Fez validação de dados?

**Classe Personagem ( linha 52 )**

*public Personagem(int id, String nome, int saude, Item[] Itens) {*

*setId(id);*

*setNome(nome);*

*setSaude(saude);*

*print();*

**Classe PersonagemMilitar ( linha 19 )**

*public PersonagemMilitar(int ataque, int defesa, int corrida,int id, String nome, int saude,Item[] Itens) {*

*super(id, nome, saude, Itens);*

*this.ataque = ataque;*

*this.defesa = defesa;*

*this.corrida = corrida;*

*}*

**Classe Legionário ( Linha 20)**

*public Legionario(int velocidadeAtaque,int ataque, int defesa, int corrida,int id, String nome, int saude,Item[] Itens) {*

*super(ataque, defesa, corrida, id, nome, saude, Itens);*

*this.velocidadeAtaque = velocidadeAtaque;*

*print();*

*}*

**Classe Item (Linha 17)**

*public Item(String item) {*

*this.item = item;*

*}*

**Classe Esquadrao (Linha 20)**

*public Esquadrao(String familia) {*

*Legionario l = new Legionario();*

*this.familia = familia;*

*qtdLegionarios++;*

*}//fecha métoco construtor.*

*public Esquadrao(){*

*this.familia = "";*

*}*

1. Dois construtores

**Classe Personagem ( linha 52 )**

*public Personagem(int id, String nome, int saude, Item[] Itens) {*

*setId(id);*

*setNome(nome);*

*setSaude(saude);*

*print();*

*public Personagem() {* ***(* Linha 63)**

*this.id = 0;*

*this.nome = "";*

*this.saude = 100;*

*print();*

*}*

**Classe PersonagemMilitar ( linha 19 )**

*public PersonagemMilitar(int ataque, int defesa, int corrida,int id, String nome, int saude,Item[] Itens) {*

*super(id, nome, saude, Itens);*

*this.ataque = ataque;*

*this.defesa = defesa;*

*this.corrida = corrida;*

*}*

*public PersonagemMilitar() {* **(linha 27)**

*super();*

*this.ataque = 0;*

*this.defesa = 0;*

*this.corrida = 0;*

*}*

**Classe Legionário ( Linha 20)**

*public Legionario(int velocidadeAtaque,int ataque, int defesa, int corrida,int id, String nome, int saude,Item[] Itens) {*

*super(ataque, defesa, corrida, id, nome, saude, Itens);*

*this.velocidadeAtaque = velocidadeAtaque;*

*print();*

*}*

*public Legionario() {* **(linha 26)**

*super();*

*this.velocidadeAtaque = 0;*

*print();*

*}*

**Classe Item (Linha 17)**

*public Item(String item) {*

*this.item = item;*

*}*

*public Item() {* **(Linha 22)**

*item = "";*

*}*

**Classe Esquadrao (Linha 20)**

*public Esquadrao(String familia) {*

*Legionario l = new Legionario();*

*this.familia = familia;*

*qtdLegionarios++;*

*}//fecha métoco construtor.*

*public Esquadrao(){*

*this.familia = "";*

*}*

*public Esquadrao(){* **(Linha 27)**

*this.familia = "";*

*}*

1. Um atributo static.

static String nome; **(Linha 45 da classe Personagem)**

1. Um atributo const static

private static final int forcaMaxima=100;  **(Linha 48 da Classe Personagem)**

private static final String tipoAtaque **( Linha 13 da classe Legionarios)**

*Correta modelagem dos statics?*

1. Um array

Item[] Itens; (**Linha 47 da classe Personagem)**

1. Método static – deve ser chamado no main
2. Sobrescrita de método: chamar dentro do método da classe. Usar o @override

*@Override*

*public void setNome(String nome){*  **(Linha 65 classe Legionário)**

*nome = "Legionário";*

*PersonagemMilitar.nome = nome;*

*System.out.println(PersonagemMilitar.nome);*

*}*

*@Override*

*public void setItens(Item[] Itens) {* **(Linha 73 classe Legionário)**

*this.Itens = new Item[3];*

*this.Itens = Itens;*

*for(int i=0; i<3;i++){*

*getItens()[i].setItem("Armamento "+i);*

*}*

*}*

1. Usar Protected acessando diretamente os atributos na classe derivada
2. Usar suas classes no main

Opcionais que garantem pontos extras:

Trabalhar com ENUM

Trabalhar com pacotes

**Acessando pelo github o diretório :** [**Legionarios**](https://github.com/ramoncordeiro/Legionarios) **/** [**Legionarios\_program**](https://github.com/ramoncordeiro/Legionarios/tree/master/Legionarios_program) **/** [**src**](https://github.com/ramoncordeiro/Legionarios/tree/master/Legionarios_program/src) **/** [**br**](https://github.com/ramoncordeiro/Legionarios/tree/master/Legionarios_program/src/br) **/ ufpa /**

**Encontrará os pacotes : Personagens, Classes Relacionadas , Main**