Personagem: A classe foi criada para criar o personagem principal do jogo. Ela possui um 9 atributos, sendo:

1 string para o name; 4 inteiros para life, defense, attack, stamina; 3 atributos(objetos) de classes próprias Inventario, Armadura e Arma; 1 vector de classe própria Habilidade;

1. Construtores:

*Personagem(std::string)* - constrói os atributos name com a string recebida e life, defense, attack e stamina em 100, 20, 20 e 100, respectivamente, por default.

*Personagem(std::string, int, int, int, int)* - constrói os atributos name, life, defense, attack e stamina com os atributos recebidos.

Os objetos Inventario, Arma, Armadura e std::vector<Habilidade> são sempre criados/inicializados com os construtores default (sem parametros).

1. Métodos do tipo get (getters), sendo:

\* 5 getters diretos, que retornam os atributos de mesmo nome:

*get\_life(), get\_defense(), get\_attack(), get\_stamina()* - retornam um inteiro;

*get\_name()* - retornam uma string;

\*4 getters relacionados aos atributos do objeto Armadura:

*get\_armor\_defense(), get\_armor\_id(), get\_armor\_price()* - retornam um inteiro;

*get\_armor\_name()* - retorna uma string;

\*4 getters relacionados aos atributos do objeto Arma:

*get\_weapon\_attack(), get\_weapon\_id(), get\_weapon\_price(*) - retornam um inteiro;

*get\_weapon\_name()* - retorna uma string;

\*3 getters relacionados aos atributos do objeto Inventario:

*get\_gold()* - retorna um inteiro;

*get\_inventory\_armor()* - retorna um vector do tipo Armadura, igual ao vetor do Inventario;

*get\_inventory\_weapon()* - retorna um vector do tipo Arma, igual ao vetor do Inventario;

\*2 getters relacionados aos atributos do vector do objeto Habilidade:

*get\_nskill()* - retorna um inteiro com o número de habilidades do vector Habilidade;

*get\_skill(int k)* – retorna a habilidade na posição k do vector Habilidade;

1. Métodos do tipo set (setters – tipo void), sendo:

\*4 setters diretos, que recebem um inteiro e mudam atributos do Personagem:

*set\_life(int), set\_defense(int), set\_attack(int), set\_stamina(int)*;

Observações:

As funções acima recebem um parâmetro *inteiro positivo*. Em caso de parâmetro negativo, elas modificam os atributos correspondentes para 0.

\*1 setter relacionado ao atributo gold do atributo Inventario:

*set\_gold(int)*;

1. Métodos para impressão de objetos (display - tipo void):

*display\_inventory()* - imprime atributos do Inventario;

*display\_armor()* - imprime atributos da armadura atual do Personagem;

*display\_weapon()* - imprime atributos da arma atual do Personagem;

*display\_skill()* - imprime as habilidades do Personagem;

*display\_stats()* - imprime os atributos name, life, defense, attack e stamina do Personagem;

1. Métodos de acesso/adição/remoção de itens no Inventario e no vector de Habilidades (tipo void):

*add\_skill(Habilidade)* - adiciona uma habilidade ao vector de Habilidade;

*remove\_skill(int y)* – remove a habilidade na posicao y do vetor de Habilidades;

*add\_armor(Armadura)* – adiciona a Armadura do parâmetro recebido ao Inventario;

*add\_weapon(Arma)* – adiciona a Arma do parâmetro recebido ao Inventario;

*remove\_armor(int id)* – remove a armadura de atributo id igual ao parametro id;

Observações: se a função não encontrar nenhum item com o id correspondente no Inventário, ela apenas mostra um erro.

*remove\_weapon(int id)* – remove a arma de atributo id igual ao parametro id.

Observações: se a função não encontrar nenhum item com o id correspondente no Inventário, ela apenas mostra um erro.

1. Métodos de atualização de arma/armadura atuais(tipo void):

*equip\_armor(Armadura)* – troca o objeto Armadura atual pelo parâmetro recebido.

*unequip\_armor()* – troca o objeto Armadura atual por um default, com atributos zerados.

*equip\_weapon(Arma)* – troca o objeto Arma atual pelo parâmetro recebido.

*unequip\_weapon()* – troca o objeto Arma atual por um default, com atributos zerados.

1. Métodos criados para uso em funções de outras classes:

*check\_gold(int price)* – retorna um booleano com valor *true* se o atributo gold do Inventário for maior ou igual ao parametro price ou *false* caso o valor seja menor.

*armor\_inventory\_position(int id)* – retorna a posição no vector de Armadura do Inventario com id correspondente.

Observações: se a função não encontrar nenhum item com o id correspondente no Inventário, ela retorna um inteiro com valor *-1*.

*weapon\_inventory\_position(int id)* – retorna a posição no vector de Arma do Inventario com id correspondente.

Observações: se a função não encontrar nenhum item com o id correspondente no Inventário, ela retorna um inteiro com valor *-1*.