

COM220 – Programação Orientada a Objetos

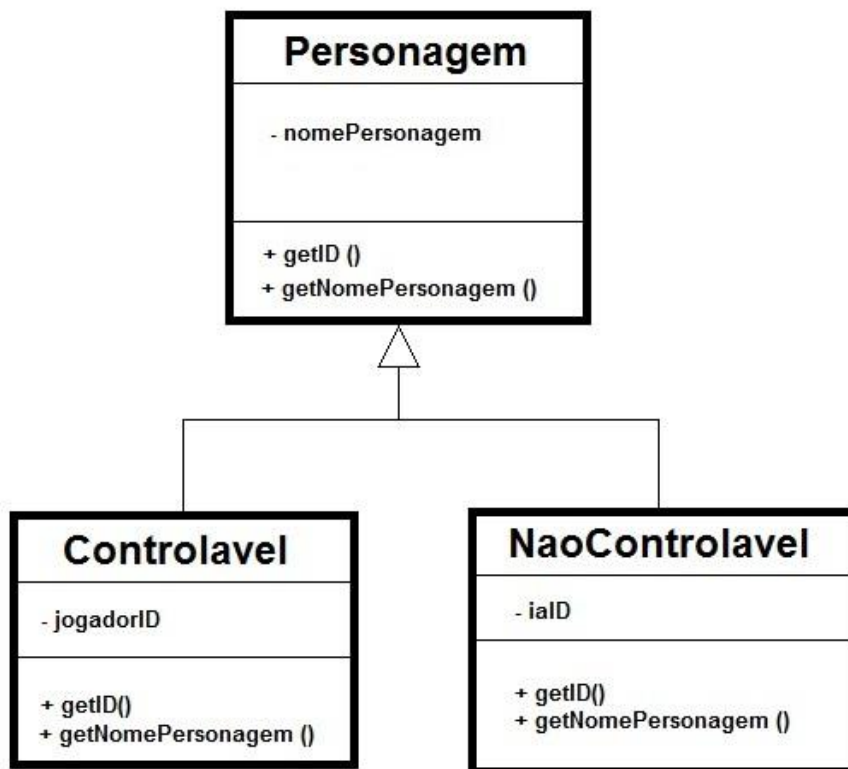
Exercício 02 – Entrega: 17/04

Um jogo de RPG (Role-Playing Game), conhecido no português como jogo de interpretação de personagens, é composto, obviamente, por personagens, os quais podem ser classificados em dois tipos:

- NPCs (Non-Player Characters): personagens não controláveis por jogadores. São aqueles personagens controlados por algum tipo de inteligência artificial ou que têm funções mais específicas dentro do jogo, como vender ou comprar itens para os jogadores.

- PCs (Player Characters): personagens controlados por jogadores específicos. Se você for jogar um RPG destes em seu computador você somente poderá criar e jogar com um personagem do tipo controlável (PC).

Tendo este tipo de hierarquia em mente você deve modificar ambos o **programa** e o **diagrama UML** dados abaixo usando os conceitos de HERANÇA, POLIMORFISMO e ABSTRAÇÃO:



Observações:

- `nomePersonagem` é o nome do personagem a ser controlado ou não. Exemplo: "Grul, o Bárbaro"
- `getID()` é o método que deve ter apenas sua definição na classe **Personagem** e devolver o ID apropriado para cada um dos tipos de personagem.
- `getNomePersonagem()` devolve o valor apropriado para tipo de personagem:
Personagem - return (`nomePersonagem`);
Controlavel - return ("Personagem Controlável " + `nomePersonagem`);
NaoControlavel - return ("Personagem Não Controlável " + `nomePersonagem`);
- `JogadorID` é o nome do jogador que controla o personagem em questão. Exemplo: "Pedro"
- `iaID` é o tipo de Inteligencia artificial que controla aquele personagem. Exemplo: "Vendedor de Poções"

O programa a ser corrigido é apresentado a seguir juntamente com uma classe main para testar suas funções. É importante notar que, ao corrigir o programa usando abstrações (classe abstrata), a terceira instância de Personagem irá apresentar um erro de compilação por não ter um tipo definido e deverá ser removida. Após verificar o erro, escreva um novo método main utilizando um ArrayList. Você deve colocar as instâncias das classes Controlavel e NaoControlavel no ArrayList e processar o ArrayList utilizando polimorfismo.

Códigos:

```
// Define a super classe Personagem
public class Personagem {

    private String nomePersonagem;

    // Construtor
    public Personagem (String nomePersonagem) {
        this.nomePersonagem = nomePersonagem;
    }

    public String getNomePersonagem() {
        return nomePersonagem;
    }

    public String getID() {
        System.out.println("ERRO: Tipo desconhecido! Nao podemos definir o ID!!!");
        return "";
    }
}
```

```
// Define a classe de personagem CONTROLAVEL, subclasse de Personagem
public class Controlavel extends Personagem {
    private String jogadorID;

    // Constructor
    public Controlavel(String nomePersonagem, String jogadorID) {
        super(nomePersonagem);
        this.jogadorID = jogadorID;
    }

    @Override
    public String getNomePersonagem() {
        return "Personagem Controlável "+super.getNomePersonagem();
    }

    @Override
    public String getID() {
        return "Nome do Jogador: "+jogadorID;
    }
}
```

```
// Define a classe de personagem NAO CONTROLAVEL, subclasse de Personagem
public class NaoControlavel extends Personagem {
    private String iaID;

    // Constructor
    public NaoControlavel(String nomePersonagem, String iaID) {
        super(nomePersonagem);
        this.iaID = iaID;
    }
}
```

```
@Override
public String getNomePersonagem() {
    return "Personagem NÃO Controlável "+super.getNomePersonagem();
}

@Override
public String getID() {
    return "Inteligencia Artifical do Tipo: "+iaID;
}
}
```

// Uma classe de teste para a classe personagem e suas subclasses

```
public class TestaPersonagem {
    public static void main(String[] args) {

        //Instanciando e imprimindo o personagem 1, um personagem Controlavel
        Personagem personagem1 = new Controlavel("'Grul, o Bárbaro'", "Pedro");
        System.out.println("Nome: "+personagem1.getNomePersonagem());
        System.out.println(personagem1.getID());

        System.out.println();

        //Instanciando e imprimindo o personagem 2, um personagem nao Controlavel
        Personagem personagem2 = new NaoControlavel("'Adria'", "Vendedora de Pocios");
        System.out.println("Nome: "+personagem2.getNomePersonagem());
        System.out.println(personagem2.getID());

        System.out.println();

        // Construir uma instancia de Personagem sem um tipo apresenta problemas
        Personagem personagemGenerico = new Personagem("Cort, o Arqueiro");
        System.out.println("Nome: "+personagemGenerico.getNomePersonagem());
        System.out.println(personagemGenerico.getID());
    }
}
```
