Inatel
Instituto Nacional de Telecomunicações

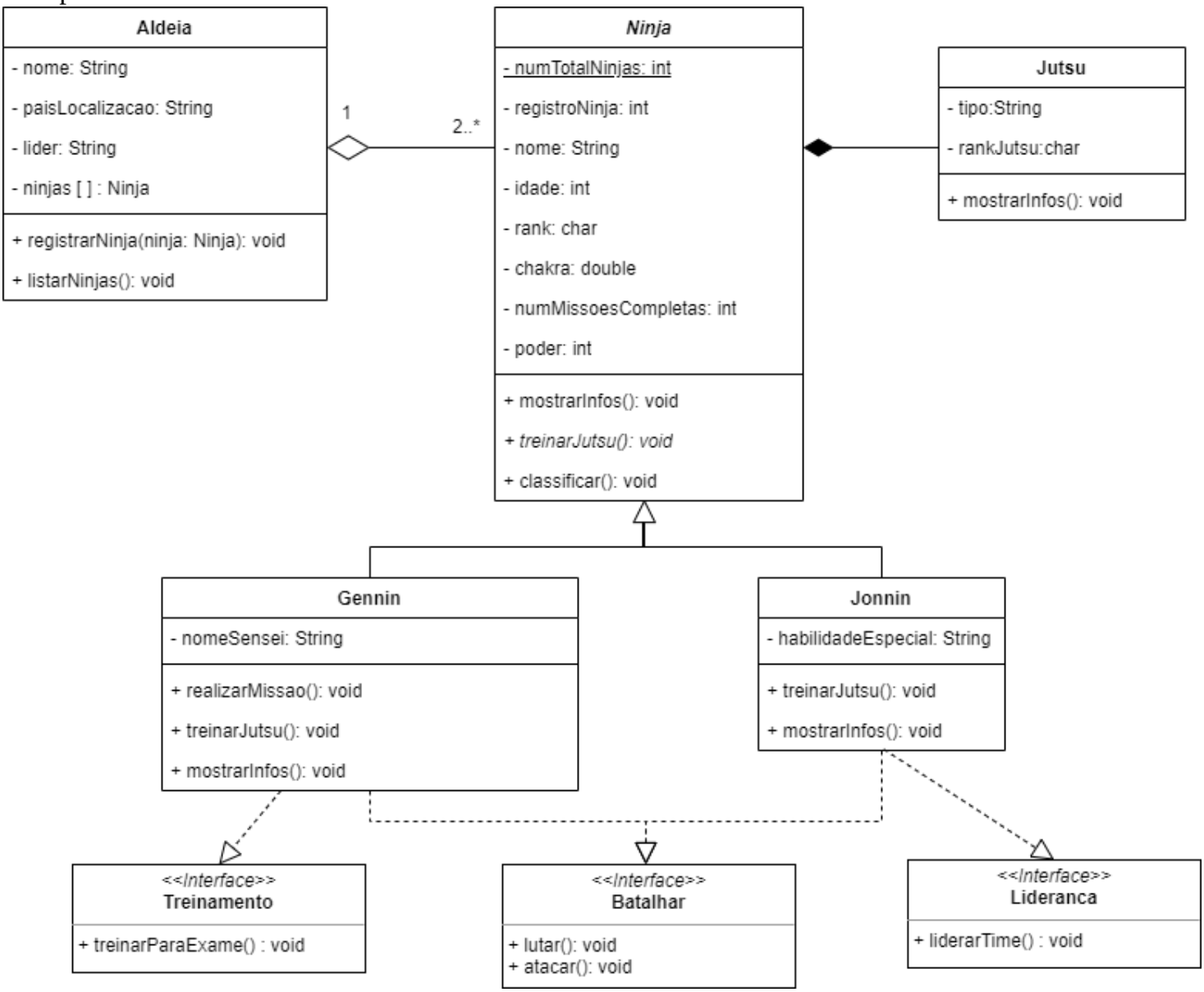
C206 – Programação Orientada a Objetos – 2023/2

Professor: Christopher de Souza Lima Francisco

Monitor: Gabriel Ilian Fonseca Barboza

Segunda Avaliação de C206 – L4

Naruto é uma das séries de anime mais icônicas do mundo, com uma vasta base de fãs e um universo complexo. Recentemente, devido ao seu sucesso, um jovem entusiasta da série solicitou que você desenvolvesse uma aplicação para facilitar a explicação sobre o funcionamento da trama de Naruto. Abaixo, é destacado um diagrama UML simplificado que representa o funcionamento do anime e suas particularidades.



O jovem destacou para você que a aplicação deve funcionar de acordo com as especificações que ele te passou, logo você deve seguir a UML que ele especificou. Na classe principal, ou seja, na Main crie pelo menos um Ninja de cada tipo (Gennin, Chunnin e Jonnin), preencha todos seus atributos e adicione à lista de Ninjas através do método adicionarNinja (). Execute todos os métodos de cada personagem e ao final mostre a informação de todos eles.

Notas Importantes:

- A classe Ninja deve ser abstrata e o atributo numTotalNinjas do tipo estático;
- Adicione os Ninjas a aldeia através do método registrarNinja();
- Incremente o contador para cada objeto criado e use para preencher o registroNinja no construtor;

- Ao chamar o método “classificar()” do ninja faça um controle de fluxo para classificar o rank daquele determinado Ninja. O ranqueamento dos Ninjas é feito através do número de missões completas por ele:
    - A – 21 a 40 missões, são rank A.
    - B – 11 a 20 missões, são rank B.
    - C – 2 a 10 missões, são rank C.
    - S – 40 ou mais missões, são rank S

  No controle de fluxo você deverá, portanto, alterar a sua variável **rank** para o ranqueamento do Ninja de acordo com o número de missões completas. Lembre-se que o único método que deve ser usado para modificar essa variável é o método “classificar()”, sendo assim é necessário chama-lo antes de mostrar as informações do Ninja e que ninjas com menos de 2 missões ainda não tem raqueamento.
- No método “lutar ()” implementado através da interface “Batalhar” faça um sout “Está acontecendo uma luta” e em seguida mostre todas as informações do ninja que está participando da luta.
- No método “atacar()” implementado através da interface “Batalhar” faça um controle de fluxo para atender as especificações de acordo com a habilidade especial do ninja caso ele seja um **Jonnin**.
    - Clones– Decremento de 5% do chakra do Ninja.
    - Teletransporte – Decremento de 7% do chakra do Ninja.

  No controle de fluxo você deverá, portanto, alterar o nível de chakra durante um ataque daquele Jonnin, isso é classificado pela habilidade especial do Ninja, como demonstrado anteriormente. Caso ele seja um **Gennin** envie somente a mensagem “Ataque fraco, sem chakra utilizado”. Em ambos o caso mostre o nome e o chakra do ninja após o ataque.
- No método “mostrarInfos()” herdado da classe mãe Ninja, apresente todas as informações do ninja, bem como do seu jutsu.
- No método “treinarJutsu()” herdado da classe mãe Ninja, se o Ninja for um Gennin e possuir um Jutsu do tipo Kekkei-Genkai faça um incremento de 15 unidades no **poder** dele, se o jutsu for do tipo taijutsu faça um incremento de 5. Agora se ele for um Jonnin e possuir um jutsu do tipo Kekkei-Genkai faça um incremento de 100, se for taijutsu o incremento é de 40. Ao final do treinamento apresente uma mensagem constando o nome do ninja seguido do novo poder dele. Exemplo : “ O Naruto realizou um treinamento do seu jutsu e agora seu poder é 120”.
- No método “realizarMissao()” da subclasse Gennin faça apenas um sout informando o nome do ninja seguido de “ está realizando uma missão “.
- No método “treinarParaExame()” implementado atráves da interface “Treinamento” na subclasse Gennin se o nome do sensei do Gennin for “Kakashi” faça um incremento da força de 10 unidades, e faça um sout com a mensagem “O ninja é treinado pelo Kakashi, logo vai ter mais poder”. Se o sensei for outro, apenas imprima a mensagem com o nome do ninja seguido de “ está em treinamento”
- **Ao final do código mostre a quantidade de ninjas que registrados na aldeia naquele momento**.
- Não é necessário fazer entrada de dados , todos os atributos deverão ser preenchidos na classe Main de forma estática, lembre-se também que a única forma para preencher a variável ‘rank’ do Ninja é através do método “classificar()”.
- Preencha todos os atributos das classes, não esqueça dos Getters e Setters para os atributos do tipo private!!
- Ao reescrever o método mostraInfos() nas classes filhas, chame o método já criado na classe mãe através do atributo super.mostraInfo() para poupar código.
- AVALIE corretamente com relação aos getters e setters.

Dicas:

- Para implementar mais de uma interface, usa-se a vírgula. Exemplo: implements Interface1, Interface2

- Ao salvar um Ninja no array, use um break para evitar que todas as posições do array sejam preenchidas com o mesmo valor;
- Tente executar todos os métodos do Ninja dentro do método listarNinjas.