MAC 110 – Introdução à Ciência da Computação para BCC
IME – Primeiro Semestre de 2009

Primeiro Exercício-Programa — Data de entrega: até 17 de abril de 2009

Professor: Marcelo Finger

Jogo do NIM

Neste jogo, n peças são inicialmente dispostas numa mesa. Dois jogadores jogam alternadamente, retirando pelo menos 1 e no máximo m peças cada um. Quem tirar as últimas peças possíveis ganha o jogo.

Existe uma uma estratégia para ganhar o jogo e é muito simples, consiste em deixar sempre múltiplos de $m + 1$ peças ao jogador oponente.

1 Objetivo

Você deverá utilizar o DrJava para construir uma classe que permita a uma "vítima"jogar o NIM contra o computador. O computador, é claro, deverá seguir a estratégia vencedora descrita acima.

Seja n o número de peças inicial e m o número máximo de peças que é possível retirar em uma rodada. Então há dois cenários possíveis para o inicio do jogo:

- Se n é múltiplo de $m+1$, o computador deve ser "generoso" e convidar o jogador a iniciar a partida.
- Caso contrário, o computador toma a inciativa de começar o jogo.

Uma vez iniciado o jogo, a estratégia do computador para ganhar consiste em deixar sempre um número de peças que seja múltiplo de $m+1$ ao jogador. Caso não seja possível, deverá tirar o número máximo de peças possíveis.

Seu trabalho será implementar o Jogo e que o computador faça uso da estratégia ganhadora.

2 Seu Programa

Com o objetivo do EP já definido, uma dúvida que deve surgir nesse momento é como modelar o jogo de forma que possa ser implementado como uma classe em Java que corresponda rigorosamente às especificações descritas até agora.

Felizmente, para facilitar a vida de todos, sugerimos, a seguir, um possível modelo, isto é, uma descrição em linhas gerais de uma classe que resolve o problema proposto neste EP. Embora não seja estritamente obrigatório, recomendamos fortemente que você siga o modelo abaixo neste seu primeiro exercício-programa em Java.

- Uma única classe `JogoNim`.
- Atributos:
  - `númeroPeçasAtual`: Inteiro que representa o número de peças em jogo. Seu valor será passado ao iniciar o jogo.
  - `o maxPeçasPermitidoRemover`: Inteiro que representa o número máximo de peças que pode ser retirado em cada rodada.
- Métodos para interação com o usuário
  - `void iniciaPartida(int númeroPeças, int númeroMáximo)`. Inicia uma nova partida. Recebe como parâmetros o número inicial de peças e o número máximo possível de peças a retirar.
  - `void novaRodada( int peçasRetiradas )`

    Executa uma rodada completa, isto é, recebe o número `peçasRetiradas` que representa a quantidade de peças retiradas pelo jogador humano; em seguida, verifica se a jogada é válida (se não for, emite uma mensagem para o usuário jogar novamente). A seguir, verifica se o jogador retirou o último peça; caso contrário, executa a jogada do computador (com a estratégia vencedora), informando ao usuário (imprimindo na tela) quantas peças o computador retirou e quantas sobraram; e por último verifica se o jogo acabou.

3 Execução

Veja um exemplo de como deve funcionar a simulação na janela do interpretador do DrJava:

```
Welcome to DrJava.
> JogoNim jogo = new JogoNim();
> jogo.iniciaPartida( 16, 3 );
Tenha a bondade de iniciar.
Restam 16 peças, faça sua jogada.
> jogo.novaRodada( 2 );
Restam 14 peças.
Restirei 2 peças.
Restam 12 peças, faça sua jogada.
> jogo.novaRodada( 1 );
Restam 11 peças.
Restirei 3 peças.
Restam 8 peças, faça sua jogada.
> jogo.novaRodada( 3 );
Restam 5 peças.
Restirei 1 peças.
Restam 4 peças, faça sua jogada.
> jogo.novaRodada( 1 );
Restam 3 peças.
Restirei 3 peças.
Ganhei!!!.  Fim da partida.
```

Para facilitar mais ainda a vida dos alunos (como estamos bonzinhos hoje, não?), sugerimos a seguir alguns métodos auxiliares que podem ser utilizados pelos próprios métodos da classe `JogoNim` descritos acima. Esses métodos melhorariam a organização do código da sua classe.

Métodos de uso interno da própria classe (não são obrigatórios, e você pode modificar seus parâmetros da forma que achar mais conveniente para implementar a sua estratégia de jogo):

- `void imprimePeças()`: Imprime o número de peças em jogo (quantos estão sobrando).
- `void jogadaDoComputador()`: Executa a jogada do computador segundo a estratégia vencedora e, a seguir, imprime o número de peças retirados pelo computador.
- `void verificaFimDeJogo()`: Verifica se o jogo acabou, isto é, se o computador já venceu o jogo. Em caso afirmativo, imprime uma mensagem Caso contrário, avisa que é a vez do jogador humano.

3.1 Apenas para alunos do BCC

Uma vez que a classe `JogoNim` esteja funcionando, você deverá (mo mesmo arquivo) criar uma outra classe `CampeonatoJogoNim`, copiando todos os atributos e métodos da classe `JogoNim` para o seu interior. Em seguida, à sua classe deverão ser adicionados novos atributos e métodos que computem um campeonato de partidas de Jogo NIM, contendo o seguintes métodos:

- `void iniciaCampeonato()`. que marca o início de um campeonato. Uma execução pode ter várias partidas antes de um campeonato (de treino) e até partidas entre campeonatos.

  Um campeonato pode ter quantas partidas vocês quiserem, e cada uma delas deve ser realizada exatamente como antes: iniciada por `iniciaPartida( nPeças, maxRetiradas)` e `novaRodada( nRetiradas)`.
- `void imprimePlacarCampeonato()`. Mostra o número de vitórias do computador e do jogador. **Nota: algumas falhas na programação podem levar o jogador a VENCER partidas, mesmo usando da interface oficial do programa.** Cuidado! Testaremos estas possibilidades e ver como o seu programa reage.
- `void fimDeCampeonato()`. Encerra um campeonato e imprime o placar e o vencedor.

4 Observações importantes

4.1 Sobre a elaboração:

Este EP pode ser elaborado por equipes de dois alunos, desde que sejam respeitadas as seguintes regras.

- Os alunos devem trabalhar sempre juntos. A idéia é que deve existir uma cooperação;
- Caso em um grupo exista um aluno com maior facilidade, este deve explicar as decisões tomadas. E o seu par deve participar e se esforçar para entender o desenvolvimento do programa (Chamamos isso de *programação em pares*, que é uma excelente prática que vocês devem se esforçar para adotar);
- Mesmo a digitação do EP deve ser feita em grupo, enquanto um digita, o outro fica acompanhando o trabalho;
- Recomendamos fortemente que o exercício seja desenvolvido da forma descrita na observação acima, mas também pode ser feito individualmente.

4.2 Sobre a avaliação:

- É sua responsabilidade manter o código do seu EP em sigilo, ou seja, apenas você e seu par devem ter acesso ao código.
- No caso de exercícios feitos em dupla, a mesma nota da correção será atribuída aos dois alunos do grupo;
- **Não serão toleradas cópias!** Exercícios copiados (com ou sem eventuais disfarces) levarão à reprovação da disciplina e o encaminhamento do caso para a Comissão de Graduação do aluno.
- Exercícios atrasados não serão aceitos;
- Exercícios com erros de sintaxe (ou seja, erros de compilação) receberão nota ZERO;
- É muito importante que seu programa tenha comentários e esteja bem indentado, ou seja, digitado de maneira a ressaltar a estrutura de subordinação dos comandos do programa (conforme visto em aula). A qualidade do seu trabalho sob esse ponto de vista influenciará sua nota!
- As informações impressas pelo seu programa na tela devem aparecer da forma mais clara possível. Este aspecto também será levado em consideração no cálculo da sua nota;
- Uma regra básica é a seguinte: do ponto de vista do monitor responsável pela correção dos trabalhos, quanto mais convenientemente apresentado estiver o seu programa, melhor avaliado será seu trabalho.

4.3 Sobre a entrega:

- O prazo de entrega é o dia 17/04/2009;
- Entregar apenas um arquivo de nome `JogoNim.java`; a classe `CampeonatoJogoNim` deve estar ao fim deste arquivo também.
- No início do arquivo, acrescente um cabeçalho bem informativo, como o seguinte:

```
/*****************************************************************/
/**   MAC 115 - Introdução à Computação                         **/
/**   IME-USP - Primeiro Semestre de 2009                       **/
/**   <turma> - <nome do professor>                             **/
/**                                                             **/
/**   Primeiro Exercício-Programa  --  jogo do NIM              **/
/**   Arquivo: JogoNim.java                                     **/
/**                                                             **/
/**   <nome do(a) aluno(a)>               <número USP>          **/
/**   <nome do(a) aluno(a)>               <número USP>          **/
/**                                                             **/
/**   <data de entrega>                                         **/
/*****************************************************************/
```

Não é obrigatório que o cabeçalho seja idêntico a esse, apenas que contenha pelo menos as mesmas informações.

- Para a entrega, utilize o Paca. Você pode entregar várias versões de um mesmo EP até o prazo, mas somente a última será armazenada pelo sistema.
- Não serão aceitas submissões por email ou atrasadas. Não deixe para a última hora, pois o sistema pode ficar congestionado e você poderá não conseguir enviar.
- Guarde uma cópia do seu EP pelo menos até o fim do semestre!