



Modelo Executável

CONTEÚDO

2.1 Modelo Executável de uma Aplicação	1
2.2 Sobre a Avaliação	3

2.1 Modelo Executável de uma Aplicação

Como parte do processo de desenvolvimento de uma aplicação de software, a codificação de modelos ajuda no alinhamento entre os objetivos do usuário e o grau de entendimento do desenvolvedor.

Na aplicação *Guerra dos 30 Anos*, diversos casos de uso podem ser identificados, destacando-se:

- CDU 1: iniciar o jogo
- CDU 2: formular uma hipótese
- CDU 3: refutar uma hipótese
- CDU 4: girar a roleta
- CDU 5: seguir por uma passagem secreta

Cada um dos casos de uso impõem restrições sobre o modelo de objetos da aplicação. Assim, o modelo da aplicação referente ao caso de uso CDU 1 deve ser elaborado que tal forma que o jogo tenha início apenas quando houver de 3 a 6 jogadores.

Na tentativa de elaborar o modelo de objetos da aplicação orientado pelos casos de uso, o desenvolvedor precisa integrar os diversos modelos que foram produzidos para atender aos diferentes casos de uso. Para refutar uma hipótese (CDU 3), por exemplo, o modelo da aplicação deve ser elaborado de tal forma a obedecer a ordem de ingresso dos jogadores no jogo, conforme estabelecido no caso de uso CDU 1.

A codificação desses diferentes modelos (elaborados a partir dos casos de uso) ajuda na identificação de eventuais problemas de inconsistência da aplicação. Quando se executa o modelo codificado dentro de um cenário previamente estabelecido, os comportamentos da aplicação podem ser analisados, permitindo a verificação do modelo em elaboração.

2.2 Sobre a Avaliação

Datas Importantes

<i>Atividade AI1</i>	<i>Data</i>
Divulgação das Tarefas	15/05/2008
Entrega dos Artefatos	05/06/2008

Artefatos da Atividade AI1 Cada equipe deverá entregar um documento organizado nas seguintes partes:

- (10 pontos) **Capa do documento**, contendo:
 - (peso 20%) código da atividade,
 - (peso 20%) código da equipe (lista de matrículas e nomes dos componentes),
 - (peso 60%) data da entrega (efetiva) do documento.
- (5 pontos) **Índice do documento**, contendo: as seções e subseções indicadas nesta especificação, bem como os números das páginas onde elas ocorrem.
- (30 pontos) **CDU 2: Modelo do Caso de Uso** A equipe deverá produzir os seguintes artefatos:
 1. (peso 60%) – um texto descrevendo o fluxo primário do caso de uso CDU 2.
 2. (peso 15%) – diagrama UML do caso de uso CDU 2.
 3. (peso 25%) – diagrama UML das seqüências de mensagens ilustrando o fluxo primário do caso de uso CDU 2.
- (50 pontos) **CDU 2: Modelo de Objetos do Jogo** A equipe deverá produzir os seguintes artefatos:
 1. (peso 50%) – um texto descrevendo os objetos e as mensagens que são trocadas quando a aplicação realiza o caso de uso CDU 2. Ao longo do texto, diagramas UML de comunicações e de classes devem ser referenciados (com o peso a seguir estabelecido).
 2. (peso 20%) – diagramas UML das comunicações entre os objetos envolvidos na realização do CDU 2. Diagramas diferentes devem ser produzidos para ilustrar diferentes aspectos comportamentais.
 3. (peso 30%) – diagramas UML das classes dos objetos envolvidos na realização do CDU 2. Diagramas diferentes devem ser produzidos (i) para capturar os relacionamentos entre as classes (com papéis, multiplicidades e restrições), (ii) para enfatizar operações e respectivos atributos envolvidos nas suas implementações.
- (70 pontos) **CDU 2: Código do Modelo de Objetos do Jogo** A equipe deverá produzir os seguintes trechos de código:

1. (peso 10%) – código do método `proximoJogador()` da classe `Guerra30Anos`. Esse método deve retornar o próximo jogador a jogar, de acordo com a ordem inicial de inscrição no jogo.
2. (peso 30%) – código do método `formularHipoteseAssassinato()` da classe `Participante`. Esse método deve perguntar qual a hipótese do jogador e comunicá-la aos demais participantes, na ordem inicial de inscrição no jogo.
3. (peso 30%) – código do método `coletarEvidencias()` da classe `Guerra30Anos`. Esse método deve perguntar para cada participante, na ordem inicial de inscrição no jogo, se ele possui alguma evidência que invalide a hipótese formulada. Neste caso, o jogo deverá comunicar a evidência apenas para o jogador que formulou a hipótese, comunicando aos demais que a hipótese foi invalidada.
4. (peso 30%) – o seguinte roteiro de teste deverá ser executado sem falhas:

```
public class CDUI_FormularHipoteseAssassinato {
    public static void cenarioComunicacaoHipoteseAosParticipantes() {
        // cenário que mostra que a hipótese foi comunicada
        // a todos os participantes
        Guerra30Anos jogo = new Guerra30Anos();
        Participante participante = jogo.proximoJogador();
        participante.formularHipoteseAssassinato();
        assert jogo.participantes.foramComunicados();
    }
    public static void cenarioRefutarArma() {
        // cenário que mostra o arcabuz sendo refutado
        // por outro participante
        Guerra30Anos jogo = new Guerra30Anos();
        Participante participante = jogo.proximoJogador();
        participante.formularHipoteseAssassinato();
        jogo.coletarEvidencias();
        assert jogo.participantes.foramComunicados();
        assert participante.confirmou( Evidencia.ARCABUZ );
    }
    public static void cenarioRefutarSuspeito() {
        // cenário que mostra o Rodolfo sendo refutado
        Guerra30Anos jogo = new Guerra30Anos();
        Participante participante = jogo.proximoJogador();
        participante.formularHipoteseAssassinato();
        jogo.coletarEvidencias();
        assert jogo.participantes.foramComunicados();
        assert participante.confirmou( Evidencia.RODOLFO );
    }
    public static void cenarioRefutarLocal() {
        // cenário que mostra a Áustria sendo refutada
        // por outro participante
        Guerra30Anos jogo = new Guerra30Anos();
        Participante participante = jogo.proximoJogador();
        participante.formularHipoteseAssassinato();
        jogo.coletarEvidencias();
        assert jogo.participantes.foramComunicados();
        assert participante.confirmou( Evidencia.AUSTRIA );
    }
    public static void cenarioSemRefutacao() {
        // cenário que mostra uma hipótese sem refutação
        // por outro participante
        Guerra30Anos jogo = new Guerra30Anos();
        Participante participante = jogo.proximoJogador();
        participante.formularHipoteseAssassinato();
        jogo.coletarEvidencias();
        assert jogo.participantes.foramComunicados();
        assert participante.confirmou( Evidencia.NENHUMA );
    }
}
```

```
public static void cenarioRefutacaoFalsa1() {
    // cenário que mostra um participante com evidências
    // mas que não comunica a invalidade da hipótese formulada
    Guerra30Anos jogo = new Guerra30Anos();
    Participante participante = jogo.proximoJogador();
    participante.formularHipoteseAssassinato();
    jogo.coletarEvidencias();
    assert jogo.invalidouRefutacao() == Refutacao.ESQUECIDA;
}
public static void cenarioRefutacaoFalsa2() {
    // cenário que mostra um participante sem evidências
    // mas que comunica a invalidade da hipótese formulada
    Guerra30Anos jogo = new Guerra30Anos();
    Participante participante = jogo.proximoJogador();
    participante.formularHipoteseAssassinato();
    jogo.coletarEvidencias();
    assert jogo.invalidouRefutacao() == Refutacao.MENTIROSA;
}
}
```

Observações

1. O código do caso de uso CDU 2 deverá ser desenvolvido em um ambiente Java 6 e organizado por diretórios. O diretório `fnt\dominio` contém o código das classes `Participante`, `Guerra30Anos` (e outras pertencentes ao modelo de domínio). O diretório `casouso` deverá conter o código da classe `CDU1_FormularHipoteseAssassinato`, bem como os arquivos `compilar.bat` e `executar.bat` (para executar os métodos dos cenários do caso de uso).

A impossibilidade de compilar e executar os ensaios comportamentais provocará a perda da parcela dos pontos referente ao item *CDU 2: Código do Modelo de Objetos do Jogo*. A equipe será responsável por quaisquer ajustes nas classe do caso de uso para que seja possível a execução dos arquivos `compilar.bat` e `executar.bat`, desde que autorizada pelo professor.

2. O documento do trabalho deverá ter total aderência à estrutura estabelecida nesta especificação.
3. O documento do trabalho deve ser entregue na forma eletrônica via Moodle.
4. Quaisquer dificuldades na avaliação serão tratadas individualmente com o time pelo professor.
5. Qualquer bibliografia complementar deverá ser discriminada no *final* do documento, em uma seção com título **Bibliografia**.
6. As páginas do documento devem estar numeradas sob pena de perder 10% da nota final da atividade.
7. Eventuais consultas a colegas e professores devem ser relatadas no documento do trabalho, nas partes pertinentes. O grupo deve evitar mais de uma consulta à mesma pessoa (ou pessoas).
8. Nenhum trabalho será aceito após a data marcada para a entrega da atividade A11.