



Departamento de Informática

Mestrado em Engenharia Informática
PROVA DE TESTE PRÁTICO – Sistemas de Computação Móvel e Ubíqua
2º Semestre, 2007/2008

NOTAS: Leia com atenção cada questão antes de responder. A interpretação do enunciado de cada pergunta é um factor de avaliação do teste. **O teste é SEM consulta. A duração do teste é de 1h30.**
O enunciado contém **7** páginas que devem ser entregues com a resposta ao teste.

NOME: _____ **NÚMERO.:** _____

1) Suponha que se pretende criar uma aplicação Java 2 ME que sirva de telecomando duma televisão. Para tal, a televisão exporta um serviço Bluetooth com um identificador único conhecido da aplicação.

A aplicação a desenvolver deve permitir apenas mudar o canal da televisão. Para tal, a aplicação comunica com a televisão através de Bluetooth, usando o protocolo RFCOMM. O protocolo aplicacional cliente/servidor é o seguinte:

Telecomando -> televisão: inteiro (e.g. 3)

[muda o canal para o número indicado – 3 no exemplo]

Televisão -> telecomando: booleano (e.g. true)

[devolve true (resp. false) quando é (resp. não é) possível fazer a mudança]

A interface da aplicação consiste na lista de canais disponíveis, a ser seleccionada pelo utilizador e um comando para sair da aplicação.

No âmbito desta aplicação, responda às seguintes questões.

NOTAS: 1) ... substitui partes de código omitido por simplicidade ou por a mesmo já ter aparecido anteriormente;
2) Em cada anexo leia atentamente a sua funcionalidade e o modo como os métodos devem ser invocados nos comentários do código.

1. Complete o anexo A com o código da aplicação.
2. Complete o anexo B com o código da função *getTVId* que retorna o identificador único do serviço implementado pela televisão. Este identificador está armazenado no registo 0 do repositório de registos "tvid". Caso, por alguma razão, não seja possível obter este valor, a função deve devolver *null*.
3. Complete o anexo C com o código da função *setChannel* que efectua a comunicação com a televisão para alterar o canal. Caso seja impossível mudar o canal, a aplicação deve apresentar um alerta explicativo da situação durante 5 segundos.
4. Indique uma limitação da aplicação indicada (que não esteja fundamentalmente relacionada com a interface gráfica) e apresente uma solução para a ultrapassar.

2) Para cada pergunta, assinale como **V[erdadeira]** ou **F[alsa]** cada uma das afirmações. **As respostas erradas descontam.**

1. Relativamente a um MIDlet.
 - i. O método `stopApp` pode ser invocado pela aplicação (em resultado de alguma acção do utilizador) e pelo sistema.
 - ii. O método `pauseApp` deve libertar todos os recursos usados pela aplicação.
 - iii. A interface gráfica tem a mesma dimensão em qualquer telemóvel.
 - iv. É possível pedir ao sistema (numa aplicação) para iniciar automaticamente uma aplicação num dado momento.
2. Relativamente ao suporte multimédia disponível no Java 2 ME.
 - i. Este suporte é fornecido por um pacote adicional, disponível apenas em algumas máquinas.
 - ii. Pode permitir a reprodução de streams obtidos em tempo real a partir dum servidor na Internet.
 - iii. Ao invocar o método `start` num *Player*, este irá sempre, nesse momento, obter os recursos necessário ao início da reprodução o que impede controlar de forma fina o momento do início da reprodução.
 - iv. Ao invocar o método `close` num *Player*, deixa de ser possível utilizar esse Player.
3. Relativamente ao suporte Bluetooth disponível no Java 2 ME:
 - i. Ao procurar um serviço usando o método `searchServices`, é sempre necessário indicar o dispositivo no qual se quer efectuar a pesquisa.
 - ii. Permite obter o endereço Bluetooth de um dispositivo com o qual se comunica.
 - iii. Permite criar apenas um serviço em cada dispositivo.
 - iv. Apenas um cliente pode contactar um serviço em cada momento.
4. Relativamente ao acesso aos mecanismos de armazenamento:
 - i. Um MIDlet pode criar apenas um repositórios de registos.
 - ii. O acesso a um repositório de registos é, por omissão, uma operação que necessita de ser autorizada pelo utilizador.
 - iii. O acesso ao sistema de ficheiro é, por omissão, uma operação que necessita de ser autorizada pelo utilizador.
 - iv. O Java 2 ME não possibilita o acesso aos repositórios do sistema (e.g. lista de contactos).
5. Relativamente aos mecanismos de comunicação:
 - i. Uma conexão `HttpConnection` permite obter um recurso a partir dum servidor web (invocando um método `GET` no servidor HTTP).
 - ii. Uma conexão `HttpConnection` permite submeter um recurso num servidor web (invocando o método `PUT` ou `POST` no servidor HTTP).
 - iii. Um MIDlet pode criar um servidor TCP, mas os clientes têm de ser atendidos sequencialmente.
 - iv. O suporte web services do Java 2 ME apenas suporta o contacto com web services desenhados especificamente para serem acedidos por dispositivos móveis (à semelhança do WAP relativamente ao HTTP).

- 3) Considere o exemplo do snake/pong multi-utilizador desenvolvido durante as primeiras aulas, em que identificamos por TelA o telemóvel do utilizador A e TelB o telemóvel do utilizador B.

Explique brevemente como poderia manter o estado do jogo com dois utilizadores, garantindo que ambos os jogadores observam o mesmo estado do jogo (e o mesmo efeito das acções dos utilizadores).

- 4) Relativamente ao trabalho prático, responda a **UMA** das seguintes perguntas (nota: caso responda a mais de uma pergunta, apenas a primeira será corrigida).

1. Considere o trabalho "Nova Interactive Boards". Suponha que é possível, no computador fixo que controla o écran partilhado, obter a intensidade do sinal de Bluetooth dos vários dispositivos portáteis ao longo do tempo. Explique como poderia usar esta informação na decisão de qual a informação a apresentar no écran.

2. Considere o trabalho "Nova Social Net". É possível implementar o trabalho criando apenas um serviço Bluetooth a executar nos dispositivos móveis ou nas gateways. Apresente, justificando, uma vantagem de cada uma das soluções.

3. Considere o trabalho "Nova Orchestra". Explique, **de forma breve mas completa**, como pode garantir que todos os telemóveis começam a reproduzir as músicas exactamente no mesmo momento.

ANEXO A

```
/** Classe principal da aplicação */
public class Telecomando extends 
{
    private Display mDisplay;
    private List mainMenu;
    private  exitCmd;

    public Telecomando() {
        init();
    }

    /** Devolve array com lista de canais disponíveis */
    private String[] getChannels() { ... }

    /** Função que cria interface gráfico */
    private void init() {

        mainMenu = new List( "Menu canais", List.IMPLICIT);
        // adiciona canais ao menu



        // adiciona controlo para sair da aplicação
        exitCmd = new  ( "Exit", ..., ...);


        // adiciona controlo para sair da aplicação
        mainMenu.setCommandListener( new CommandListener() {
            public void commandAction( Command cmd, Displayable arg1) {
                if( cmd == List.SELECT_COMMAND) {

                }
            }
        });
    }

    /** Comunica com a televisão para alterar o canal.
     * Caso seja impossível mudar o canal deve apresentar um alerta no
     * telemóvel, o qual deve ser apresentado durante 5 segundos.
     */
    protected void setChannel( int ch) { ... }
```

```

protected void  ( boolean arg0)
    throws MIDletStateException {
    notifyDestroyed();
}

protected void () {

}

protected void () throws MIDletStateException {
    mDisplay = Display.getDisplay( this);
    showMainMenu();
}

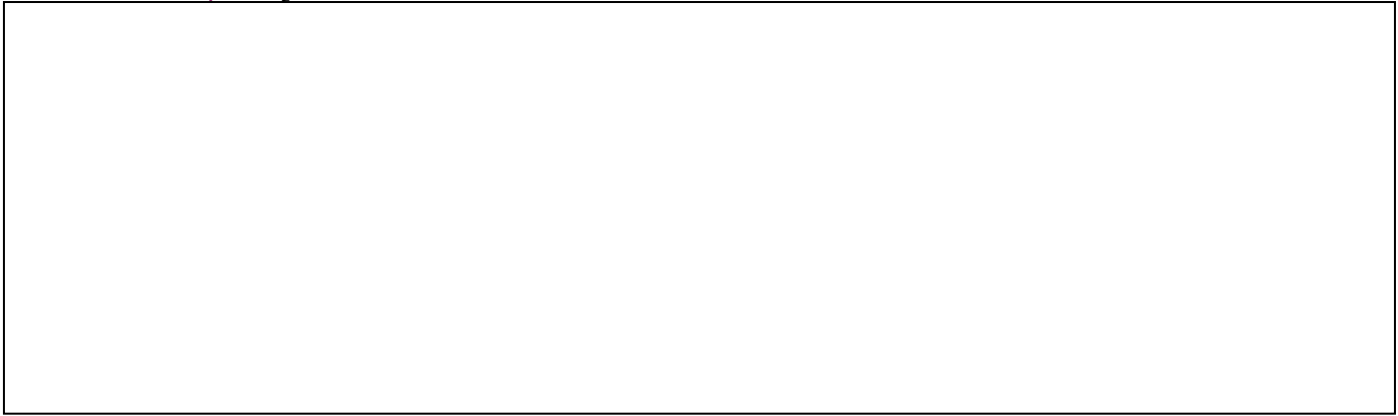
void showMainMenu() {

}

}

```

ANEXO B

```
/** Classe principal da aplicação */
public class Telecomando extends ...
{
    /** Devolve identificador único do serviço Bluetooth implementado
     * pela TV.
     * Este identificador está armazenado no registo 0 do repositório de
     * registos "tvid".
     * Deve devolver null caso não seja possível obter o registo.
     */
    protected String getTVId() {
        
    }
}
```

ANEXO C

```
/** Classe principal da aplicação */
public class Telecomando extends ...
{
    /** Devolve identificador único do serviço Bluetooth implementado
     * pela TV.
     * Este identificador está armazenado no registo 0 do repositório de
     * registos "tvid".
     * Deve devolver null caso não seja possível obter o registo.
     */
    protected String getTVId() { ... }

    /** Comunica com a televisão para alterar o canal.
     * Caso seja impossível mudar o canal deve apresentar um alerta no
     * telemóvel, o qual deve ser apresentado durante 5 segundos.
     */
    protected void setChannel( int ch) {

}

}

class BluetoothDiscovery {
    /** retorna URL para uma televisão que esteja nas imediações e
     * implemente o serviço identificado pelo identificador tvid */
    static String getTVURL( String tvid) { ... }
}
```