— Exame —
Desenvolvimento de Sistemas de Informação

LESI/LMCC
Chamada 1 - 2004/05

15/06/2005

Duração máxima: 2h00
Leia o exame com atenção.

Grupo I

Considere o seguinte excerto de código Java:

```
public class JTurma2 extends JFrame implements Observer {
    private Turma turma;
    ...
}

public class Turma extends Observable {
    private Map<String,Aluno> turma;

    public Turma() {
        this.turma = new TabAl();
    }

    public void addAluno(Aluno a) { ... }

    public Aluno getAluno(String num) throws TurmaException { ... }

    public void delAluno(String num) throws TurmaException {
        if (!this.turma.containsKey(num)) {
            StringBuffer sb = new StringBuffer("Aluno ");
            sb.append(num);
            sb.append(" inexistente!");
            throw new TurmaException(sb.toString());
        }
        this.turma.remove(num);
        this.setChanged();
        this.notifyObservers();
    }

    public int quantosPassam() {
        Collection<Aluno> col = this.turma.values();
        int tot = 0;

        for (Aluno a: col)
            if (a.getMedia()>=10)
                tot++;
        return tot;
    }
    ...
}

public class TabAl implements Map<String,Aluno> {
    public Connection conn;
```

```
    public TabAl () { ... }

    public void clear () { ... }

    public boolean containsKey(Object key) throws NullPointerException {
        try {
            Statement stm = conn.createStatement();
            String sql = "SELECT nome FROM TAlunos WHERE numero='"+(String)key+"'";
            ResultSet rs = stm.executeQuery(sql);
            return rs.next();
        }
        catch (Exception e) {throw new NullPointerException(e.getMessage());}
    }

    public Aluno get(Object key) { ... }

    public Aluno put(String key, Aluno value) { ... }

    public Aluno remove(Object key) {
        try {
            Aluno al = this.get(key);
            Statement stm = conn.createStatement();
            String sql = "DELETE '"+key+"' FROM TAlunos";
            int i  = stm.executeUpdate(sql);
            return al;
        }
        catch (Exception e) {throw new NullPointerException(e.getMessage());}
    }

    public Collection<Aluno> values() {
        try {
            Aluno a;
            String num, nome;
            int notaT, notaP;
            Collection<Aluno> col = new HashSet<Aluno>();
            Statement stm = conn.createStatement();
            ResultSet rs = stm.executeQuery("SELECT * FROM TAlunos");
            while (rs.next()) {
                num = rs.getString(1);
                nome = rs.getString(2);
                notaT = rs.getInt(3);
                notaP = rs.getInt(4);
                a = new Aluno(num,nome,notaT,notaP);
                col.add(a);
            }
            return col;
        }
        catch (Exception e) {throw new NullPointerException(e.getMessage());}
    }
    ...
}

public class Aluno {
    private String nome;
    private String numero;
    private int notaT, notaP;
    ...
}
```

1. Construa um **Diagrama de Classes** para o código apresentado (inclua no modelo o máximo de informação possível).

2. Desenhe um **Diagrama de Sequência** para o método `void delAluno(String num)` da classe `Turma`.

Grupo II

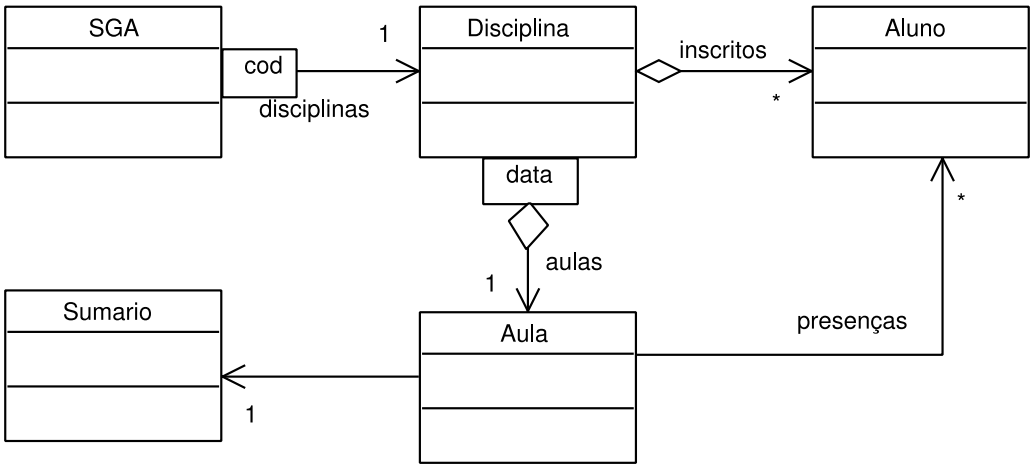
Relembre o trabalho prático:

> Pretende-se agora adicionar ao sistema desenvolvido um módulo para gestão de aulas. Para cada aula o docente deverá poder preencher o sumário e os alunos assinalar a sua presença. Por questões de segurança a identidade do docente deverá ser verificada antes de os sumários poderem ser preenchidos. Os alunos assinalam a presença através da indicação do seu número e de uma palavra chave indicada pelo docente.
>
> O sistema a desenvolver deverá permitir a posterior edição dos sumários bem como a sua impressão e envio (por e-mail) para a secretaria. Deverá ainda permitir ao docente consultar as presenças quer por aula (que alunos estiveram numa aula) quer por aluno (a que aulas foi um aluno). Os alunos tm acesso apenas a informação relativa a eles próprios.

Responda às seguintes questões:

1. Escreva um **Diagrama de *Use Case*** para o novo módulo de gestão de aulas (para cada use case identificado indique, caso existam, as suas pré-condições).

2. Considere a proposta de arquitectura de classes para o novo módulo apresentada na figura.



Com base na arquitectura proposta, desenhe um **Diagrama de Colaboração** para o método `List<Aluno> validaPresenças(String codDisc, Date data)` (da classe SGA) que verifica se todos os alunos com presença registada numa aula estão inscritos na respectiva disciplina. O método deverá devolver uma lista com todos os alunos em situação irregular. (Considere que as tabelas são implementadas com Map e as listas com List)

Grupo III

Considere a seguinte descrição do funcionamento de um relógio com cronómetro:

- Quando é ligado o relógio mostra a hora actual (o método showtime() é utilizado para mostrar a hora).
- Se o botão mode é premido uma vez, o relógio muda para cronómetro (chamando o método showStopwatch() seguido do método clear()). Neste modo, se o botão set é premido, o cronómetro começa a contagem de tempo (chamando o método start()). Quando set é novamente premido, o cronómetro para a contagem de tempo (chamando stop()). Premir set novamente coloca o cronómetro a zero (chamando clear()).
- Se o botão mode é premido enquanto enquanto o relógio está em modo cronómetro, passa para modo de acerto de hora, ficando as horas a piscar (para isso, o método showTime() é chamado seguido de flashHours()). O botão set pode então ser premido repetidamente, o que incrementa a hora (por invocação do método incrHours()). Se o botão mode é premido novamente, o relógio passa para acerto de minutos (por invocação de flashMins()); premir o botão set incrementa os minutos (por invocação de incrMins()). Premir o botão mode uma terceira vez passa ao acerto dos segundos (é utilizado o método flashSecs()); neste modo, premir o botão set coloca os segundos a zero (chamando resetSecs()) e o relógio volta a mostrar as horas normalmente. Se o botão mode é premido em vez de set enquanto os segundos estão a piscar, o relógio volta ao modo de mostrar a hora sem alterar os segundos.

Desenhe um Diagrama de Estados que represente o funcionamento do relógio tal como descrito. Para além dos eventos que provocam as transições de estado, inclua as acções que o relógio realiza.