

F A C E — F U M E C
CIÊNCIA DA COMPUTAÇÃO
DSD II

Avaliação Final — 05 de Dezembro de 2005

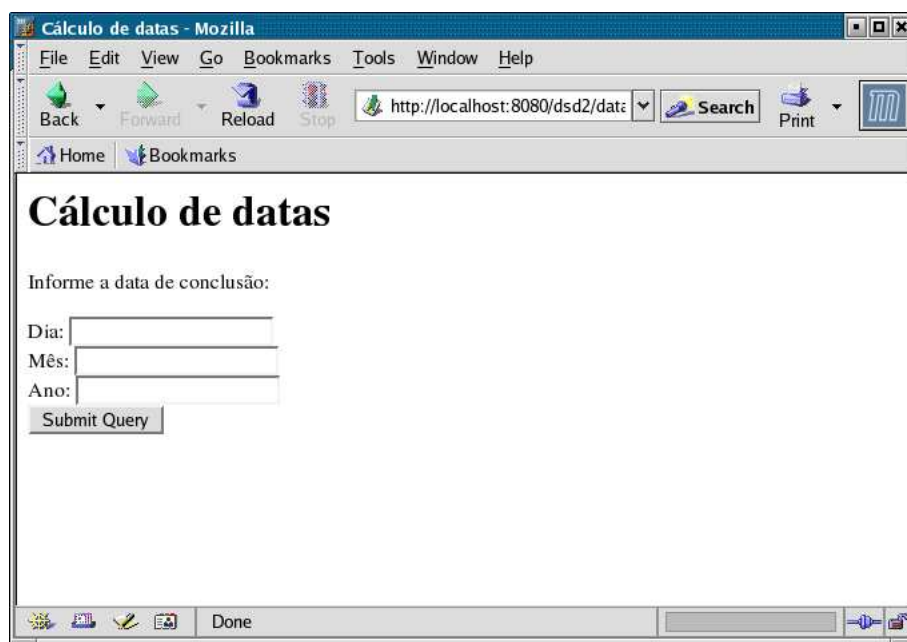
Aluno: _____

Turma: 8DA

Professor: Flávio Velloso Laper

Esta prova contém 2 questões, para um total de 35 pontos. O valor de cada questão está indicado junto da mesma. Leia atentamente todas elas antes de tentar resolvê-las

1. Uma turma de alunos está ansiosa para se formar e necessita de um servlet para fazer o cálculo do número de dias que faltam até a formatura. Construa um *servlet* denominado *DataServlet*, mapeado como */data* para fazer este cálculo. O *servlet* deve exibir uma tela como a mostrada abaixo:



A entrada de dados deve ser consistida e, caso esteja correta, o cálculo do número de dias entre a data atual e a data de conclusão deve ser feito e o resultado informado logo abaixo do botão de submissão, da seguinte forma:

Diferença: x dia(s)

onde x representa o resultado do cálculo. Se houver qualquer erro de consistência, informar:

Data inválida

Observações importantes:

1. Seu projeto deve se chamar *ProvaPrj* com contexto *prova*.
2. Você pode utilizar a classe *CalculoDatas*, mostrada abaixo, para auxiliá-lo nos cálculos. O arquivo fonte dessa classe será fornecido pelo seu professor.

```
1  import java.util.Calendar;
2  import java.util.GregorianCalendar;
3
4  public class CalculoDatas {
5
6      /**
7       * Calcula a diferenca de dias entre uma data posterior e
8       * uma anterior. Pode retornar um valor
9       * negativo se as mesmas estiverem invertidas.
10      *
11      * @param posterior Data posterior.
12      * @param anterior Data anterior.
13      * @return Diferenca de dias entre as datas.
14      */
15     public static int diferenca (GregorianCalendar posterior ,
16                                GregorianCalendar anterior) {
17         return valor(posterior) - valor(anterior);
18     }
19
20     /**
21      * Verifica se a data correspondente aos parametros recebidos
22      * e valida.
23      *
24      * @param dia Dia da data.
25      * @param mes Mes da data.
26      * @param ano Ano da data
27      * @return true se a data e valida , false caso contrario.
28      */
29     public static boolean valida(int dia , int mes , int ano) {
30         if (dia < 1 || dia > 31) return false;
31         if (mes < 1 || mes > 12) return false;
32         if (dia > 30 && (mes == 2 || mes == 4 || mes == 6 ||
33                        mes == 9 || mes == 11)) return false;
34         boolean bissexto = (ano % 400 == 0) ||
35                            (ano % 4 == 0 && ano % 100 != 0);
36         int ultDia = bissexto ? 29 : 28;
37         if (mes == 2 && dia > ultDia) return false;
38         return true;
39     }
40
41     /**
42      * Calcula um valor numerico para uma data.
43      * Este valor representa o numero de dias
44      * transcorridos entre essa data e 1/1/0.
45      */
46 }
```

```
46      * @param d Data de entrada.
47      * @return Valor numerico da data.
48      */
49      public static int valor (GregorianCalendar d) {
50          int dia = d.get(Calendar.DAY_OF_MONTH);
51          int mes = d.get(Calendar.MONTH)+1;
52          int ano = d.get(Calendar.YEAR);
53
54          if (mes < 3) {
55              ano--; mes+=9;
56          }
57          else
58              mes -=3;
59
60          int fator = ano * 365;
61          fator += (ano/4 - ano/100 + ano/400);
62          fator += (int)(Math.floor(mes*30.6 + 0.5));
63          return fator + dia + 60;
64      }
65 }
```

2. Um professor malvado resolve atrapalhar o projeto dos alunos e cria um filtro intercepta as chamadas ao *servlet* acima, fazendo com que cálculos errados sejam retornados ao chamador. O filtro retorna um valor aleatório entre 0 e 30 dias, porém não o faz em todas as chamadas. Construa o filtro para o professor seguindo as seguintes orientações:
 1. A classe do filtro deve se chamar *DataFilter*.
 2. Um valor aleatório entre 0 e 30 pode ser obtido com a instrução `(int)(Math.random()*30)`.
 3. O filtro seleciona apenas algumas chamadas para alterar. Uma chamada deverá ser alterada se o valor retornado por `Math.random()` for maior que um determinado limite; caso contrário, o retorno do *servlet* deve ser devolvido ao chamador sem qualquer modificação. O limite é um valor entre 0 e 1 que deve ser informado como um parâmetro de inicialização do filtro.
 4. A página alterada pelo filtro deve ter o mesmo formato daquela retornada pelo *servlet*.

Observações finais:

1. Observe rigorosamente as exigências sobre o nome do projeto, das classes e dos mapeamentos. Estas questões serão valorizadas.
2. Entregue sua avaliação via sistema acadêmico. Toda a árvore do projeto deve ser compactada e entregue (porém não todo o *workspace*).
3. Antes da entrega final, seu trabalho deve ser demonstrado para o professor.

BOA SORTE.