

Universidade Federal de Campina Grande – UFCG
Centro de Engenharia Elétrica e Informática – CEEI
Departamento de Sistemas e Computação – DSC

Disciplina: Laboratório de Programação 2
Período: 2011.1

Professores: Livia Campos (turma1)
Reinaldo Gomes (turma2)
Nazareno Andrade (turma 3)

Laboratório 02

Neste laboratório, experimentaremos a edição e compilação de um programa simples em Java. O laboratório tem como objetivos:

- Usar objetos de classes pré-concebidas de Java: String e Scanner.

Instruções:

Para a parte individual:

- Data de entrega: 01/03/2011 até 10h (Turmas 1 e 2) ou 12h (Turma 3);
- Salve as classes criadas em um arquivo zip chamado lab02-<seuNome>.zip (ou .tgz) e o envie para juiz.lp2@gmail.com
- Não copie o programa do seu vizinho. Se tiver dúvida converse com o professor ou com um monitor.

Para a parte em grupo:

- Data de entrega: até às 23:59h da sexta-feira 04/03 para todas as turmas.
- Salve as classes criadas em um arquivo zip chamado desafio02-<nomeDoGrupo>.zip (ou .tgz) e o envie para juiz.lp2@gmail.com
- Coloque dentro do zip ou tgz um arquivo texto com os nomes dos integrantes do grupo.

Para ambas as partes:

- A partir deste roteiro você poderá usar o Eclipse como ambiente de desenvolvimento Java;
- Crie no eclipse o projeto lab02 e programe todos os exercícios dentro do pacote lp2.lab02 que está dentro deste projeto;
- Certifique-se de que seus programas não têm erros de compilação;
- Antes da definição de cada classe escreva o seu nome em comentário (*/*Aluno: <seu nome>*/*);

Parte 1 - Individual

1. A capacidade de identificar padrões em sequências de DNA é uma operação básica na bioinformática. Implemente um programa que recebe na linha de comando uma sequência de DNA (formada por uma sequência de aminoácidos A, C, T e G), pergunta ao usuário o padrão pelo qual ele gostaria de buscar dentro da sequência e retorna a sequência com o padrão em destaque.

Como resultado o seu programa deve imprimir a sequência de entrada, com espaços removidos (os espaços acontecem apenas por descuido dos biólogos) e com o padrão, se este ocorrer, destacado em maiúsculas em cada uma de suas ocorrências. O restante da sequência deve estar em minúsculas, independente de como o usuário tenha digitado na entrada. Elimine os espaços da entrada antes de fazer o casamento de padrões (tanto nas pontas quanto no meio da sequência de entrada)!

Exemplos de execução de seu programa:

```
java lp2.lab02.BuscadorDePadroes "acgtgta"  
Informe o padrão desejado: gtg  
ocorrências: 1 - acGTGta
```

```
java lp2.lab02.BuscadorDePadroes "ActGcA ctagtacttCact gaCt"  
Informe o padrão desejado: act  
ocorrências: 5 - ACTgcACTagtACTtcACTgACT
```

```
java lp2.lab02.BuscadorDePadroes "agcatcgagcta"  
Informe o padrão desejado: cgta  
ocorrências: 0
```

O principal deste lab é que você perceba o uso de objetos da classe String. Para isso, lembre que cada String é um objeto que tem uma série de métodos que você usará na lógica do seu programa. Vários métodos podem ser úteis, dependendo da sua lógica: `toLowerCase()`, `toUpperCase()`, `split()`, `trim()`, ... dê uma estudada na API de String antes de começar a programar.

Parte 2: Desafio em grupo

1. Você deve escrever um programa para manipulação de calendário que oferece as seguintes funções:
 - a. Indicar o dia da semana (se segunda, terça, etc.) em que caiu ou cairá uma certa data. Função muito útil, especialmente para saber em que dias cairão os feriados :). Recebe como entrada três inteiros, que representam o dia, o mês e o ano em questão.
 - b. Indicar se um certo ano é ou não bissexto. Especialmente para aqueles que aniversariam no dia 29 de fevereiro é uma função bacana. O ano deve ser informado pelo usuário.

- c. Quantos dias úteis tem em um mês (excluindo, apenas, os finais de semana). Por exemplo, em fevereiro de 2011 existem 20 dias úteis. Recebe como entrada dois números inteiros representando o mês e o ano a serem considerados.

Lidar com datas não é simples. Para facilitar sua vida use objetos da classe [GregorianCalendar](#). Leiam a API da classe, pois lidar com datas é complicado e usar a classe não é trivial.

Use a seguinte representação:

Para meses:	Para dias da semana:	Para dias do mês
♦ 1 = janeiro	♦ 1 = domingo	♦ Todos os meses do ano iniciam no dia 1
♦ 2 = fevereiro	♦ 2 = segunda	♦ Alguns meses terminam no dia 30
♦ 3 = março	♦ 3 = terça	♦ Outros no dia 31
♦ 4 = abril	♦ 4 = quarta	♦ E fevereiro termina no dia 29 ou 28, dependendo de ser ou não um ano bissexto
♦ 5 = maio	♦ 5 = quinta	
♦ 6 = junho	♦ 6 = sexta	
♦ 7 = julho	♦ 7 = sábado	
♦ 8 = agosto		
♦ 9 = setembro		
♦ 10 = outubro		
♦ 11 = novembro		
♦ 12 = dezembro		

Note que na classe `GregorianCalendar` a convenção é de janeiro ser representado por 0 (zero), por exemplo. Então cuidado com esses detalhes que fazem toda a diferença.