

3 de setembro de 2018

O objetivo deste trabalho é praticar os conceitos básicos da programação orientada a objetos. O trabalho deverá ser realizado em duplas. Plágio (cópia entre diferentes grupos) não será tolerado, e grupos com trabalhos copiados (mesmo que parcialmente) receberão nota mínima no trabalho como um todo. O trabalho deverá ser realizado na primeira parte usando a linguagem de programação Java, e posteriormente na linguagem de programação C++.

A organização do seu código, clareza e detalhamento dos comentários é um critério importante de avaliação, assim como a qualidade do relatório. Discuta no relatório os testes executados (e inclua os arquivos de entrada usados para teste). A completude dos testes é um importante critério de avaliação. Declare bugs conhecidos (“*known bugs*”) no seu relatório. (Bugs conhecidos documentados são melhores do que bugs desconhecidos e não documentados.) Assim como com relação à completude dos testes a qualidade do seu relatório é um importante critério de avaliação.

O prazo para entrega da primeira versão (em Java) é dia 16 de outubro de 2018.

O prazo para entrega da segunda versão (em C++) é dia 22 de novembro de 2018.

Dado que existem várias versões dos compiladores Java e C++, fica determinado o uso das versões instaladas nas máquinas do Labgrad como versões de referência para o trabalho prático. Seu trabalho deve compilar e executar corretamente nas máquinas deste laboratório. Além disso, os arquivos de código-fonte devem estar codificados com Unicode (UTF-8) para evitar erros de compilação.

1 Contexto: Sistema Eleitoral Brasileiro

“O Sistema Eleitoral brasileiro define duas modalidades distintas de voto no País: a majoritária e a proporcional. A primeira delas é usada para escolher os chefes do executivo, ou seja, o presidente da República, os governadores e os prefeitos. Já a segunda [a proporcional] determina o modo como os representantes dos órgãos legislativos estaduais e municipais são eleitos.”

“A eleição proporcional é usada no Brasil para determinar os representantes da Câmara dos Deputados e Câmara dos Vereadores. O Sistema Eleitoral do País prevê a adoção de um sistema de lista aberta, na qual se reúne os votos gerais dos candidatos de cada partido. Tal ranking dá origem às listas partidárias, compostas pelos candidatos mais votados de determinado partido naquele pleito. Nos sistemas desse tipo, cada partido [ou coligação] obtém um número de vagas proporcionais à soma dos votos em todos os seus candidatos, e estas vagas são distribuídas, pela ordem, aos candidatos mais votados daquele partido [ou coligação].”

<http://www.brasil.gov.br/governo/2012/05/eleicao-majoritaria-e-proporcional>

“A eleição para vereador é definida através do sistema proporcional, onde primeiramente são calculados os partidos e coligações que obtiveram mais votos, e só a partir daí as vagas disponíveis em cada município são distribuídas entre os candidatos mais votados de cada partido [ou coligação].”

<https://www.eleicoes2016.com.br/como-funciona-o-sistema-proporcional/>

2 Especificação

Implemente um sistema para processar os dados disponibilizados pela Justiça Eleitoral referentes à votação de vereadores em um município das eleições de 2016.

A Justiça Eleitoral disponibiliza um sistema (“Divulga 2016”¹) em que é possível obter os resultados da votação nominal em vereadores de um município em um arquivo CSV, com o seguinte formato:

- A primeira coluna identifica um número de sequência, sendo precedido por asterisco (*) quando o candidato foi eleito, e por cerquilha (#) se o candidato não teve seus votos validados devido à sua situação jurídica ou à do seu partido.
- A segunda coluna identifica o número do candidato.
- A terceira coluna identifica a sigla do partido, e, se houver, a coligação com outros partidos. (A sigla do partido é separada da coligação com um hífen (-), e os partidos de uma coligação são separados por uma barra (/).)
- A quarta coluna identifica o número de votos recebidos pelo candidato.
- A quinta coluna indica a porcentagem dos votos válidos recebidos pelo candidato.

Eis um fragmento do arquivo gerado para a votação em Vitória, ES (primeiras 21 linhas):

```
Seq.      (i);Núm.;Candidato;Partido/Coligação;Votação;% Válidos
*0001;23123;FABRÍCIO GANDINI;PPS - PPS / PROS;7.611;4,21 %
*0002;23444;DENNINHO;PPS - PPS / PROS;6.167;3,41 %
*0003;40000;DAVI ESMAEL;PSB - PSB / PHS;5.165;2,85 %
*0004;23023;VINICIUS SIMÕES;PPS - PPS / PROS;4.403;2,43 %
*0005;45612;NEUZINHA;PSDB - PSC / PSDB;3.652;2,02 %
*0006;23000;LEONIL;PPS - PPS / PROS;3.577;1,98 %
*0007;40222;NATHAN MEDEIROS;PSB - PSB / PHS;3.316;1,83 %
*0008;55222;MAZINHO;PSD - SD / PMB / PSD / PRTB / PTC;3.090;1,71 %
*0009;14333;DALTO NEVES;PTB;3.083;1,70 %
*0010;12123;MAX DA MATA;PDT;2.339;1,29 %
*0011;20000;WANDERSON MARINHO;PSC - PSC / PSDB;2.208;1,22 %
*0012;12345;SANDRO PARRINI;PDT;2.024;1,12 %
*0013;43111;LUIZ PAULO AMORIM;PV - PP / PV / REDE;1.696;0,94 %
*0014;14222;ROBERTO MARTINS;PTB;1.547;0,86 %
*0015;11444;CLEBER FELIX (CLEBINHO);PP - PP / PV / REDE;1.524;0,84 %
0016;23222;LUIZ EMANUEL;PPS - PPS / PROS;3.138;1,73 %
0017;31444;ROGERINHO PINHEIRO;PHS - PSB / PHS;2.961;1,64 %
0018;77222;LUISINHO;SD - SD / PMB / PSD / PRTB / PTC;2.693;1,49 %
0019;23333;WAGUINHO ITO;PPS - PPS / PROS;2.087;1,15 %
0020;45000;ALOISIO VAREJAO;PSDB - PSC / PSDB;1.878;1,04 %
```

¹ <http://www.tse.jus.br/eleicoes/eleicoes-antiores/eleicoes-2016/imprensa/parceria-divulgacao-resultados-2016>

As 10 últimas linhas do arquivo são reproduzidas abaixo:

```
0238;33998;DIRCE RODRIGUES;PMN - PRP / PMN / PSL;20;0,01 %
0239;17999;CANTORA ROSE MIRIAM;PSL - PRP / PMN / PSL;8;0,01 %
0240;17001;CIRLENE MOREIRA;PSL - PRP / PMN / PSL;2;0,01 %
#0241;27256;SAUL PEREIRA;PSDC - PRB / PEN / PPL / PSDC / PC do B;0;0,00 %
#0242;12025;ZILMARIA;PDT;0;0,00 %
#0243;13680;GESSIMARA;PT;0;0,00 %
#0244;13789;MARA MATTOS;PT;0;0,00 %
#0245;23111;EDINHO BLANCHER;PPS - PPS / PROS;0;0,00 %
#0246;43210;JÉSSICA POLESE;PV - PP / PV / REDE;0;0,00 %
#0247;50000;EDMAR SANTOS;PSOL;0;0,00 %
```

Você deve criar um programa que leia os dados em um arquivo de divulgação (cujo nome é passado na linha de comando) e gere diversos relatórios na saída padrão:

- Número de vagas;
- Candidatos eleitos (sempre indicado partido, número de votos e coligação, se houver)
- Candidatos mais votados dentro do número de vagas;
- Candidatos não eleitos e que seriam eleitos se a votação fosse majoritária;
- Candidatos eleitos no sistema proporcional vigente, e que não seriam eleitos se a votação fosse majoritária, isto é, pelo número de votos apenas que um candidato recebe diretamente;
- Votos totalizados por coligação ou partido (quando um partido não estiver em coligação), número de candidatos eleitos;
- Votos totalizados por partido, número de candidatos eleitos;
- Total de votos nominais.

Em todos os casos, as saídas devem ser ordenadas por votação decrescente.

A saída deve respeitar exatamente o formato prescrito abaixo (incluindo quebras de linhas, separadores, etc.), como exemplo de execução para os dados da cidade de Vitória:

Número de vagas: 15

Vereadores eleitos:

```
1 - FABRÍCIO GANDINI (PPS, 7611 votos) - Coligação: PPS / PROS
2 - DENNINHO (PPS, 6167 votos) - Coligação: PPS / PROS
3 - DAVI ESMAEL (PSB, 5165 votos) - Coligação: PSB / PHS
4 - VINICIUS SIMÕES (PPS, 4403 votos) - Coligação: PPS / PROS
5 - NEUZINHA (PSDB, 3652 votos) - Coligação: PSC / PSDB
6 - LEONIL (PPS, 3577 votos) - Coligação: PPS / PROS
7 - NATHAN MEDEIROS (PSB, 3316 votos) - Coligação: PSB / PHS
8 - MAZINHO (PSD, 3090 votos) - Coligação: SD / PMB / PSD / PRTB / PTC
9 - DALTO NEVES (PTB, 3083 votos)
10 - MAX DA MATA (PDT, 2339 votos)
11 - WANDERSON MARINHO (PSC, 2208 votos) - Coligação: PSC / PSDB
12 - SANDRO PARRINI (PDT, 2024 votos)
13 - LUIZ PAULO AMORIM (PV, 1696 votos) - Coligação: PP / PV / REDE
14 - ROBERTO MARTINS (PTB, 1547 votos)
15 - CLEBER FELIX (CLEBINHO) (PP, 1524 votos) - Coligação: PP / PV / REDE
```

Candidatos mais votados (em ordem decrescente de votação e respeitando número de vagas):

- 1 - FABRÍCIO GANDINI (PPS, 7611 votos) - Coligação: PPS / PROS
- 2 - DENNINHO (PPS, 6167 votos) - Coligação: PPS / PROS
- 3 - DAVI ESMAEL (PSB, 5165 votos) - Coligação: PSB / PHS
- 4 - VINICIUS SIMÕES (PPS, 4403 votos) - Coligação: PPS / PROS
- 5 - CAMILA VALADÃO (PSOL, 3727 votos)
- 6 - NEUZINHA (PSDB, 3652 votos) - Coligação: PSC / PSDB
- 7 - ZEZITO MAIO (PMDB, 3590 votos)
- 8 - LEONIL (PPS, 3577 votos) - Coligação: PPS / PROS
- 9 - NATHAN MEDEIROS (PSB, 3316 votos) - Coligação: PSB / PHS
- 10 - LUIZ EMANUEL (PPS, 3138 votos) - Coligação: PPS / PROS
- 11 - MAZINHO (PSD, 3090 votos) - Coligação: SD / PMB / PSD / PRTB / PTC
- 12 - DALTO NEVES (PTB, 3083 votos)
- 13 - ROGERINHO PINHEIRO (PHS, 2961 votos) - Coligação: PSB / PHS
- 14 - DEVANIR FERREIRA (PRB, 2948 votos) - Coligação: PRB / PEN / PPL / PSDC / PC do B
- 15 - MAURICIO LEITE (PMDB, 2864 votos)

Teriam sido eleitos se a votação fosse majoritária, e não foram eleitos:
(com sua posição no ranking de mais votados)

- 5 - CAMILA VALADÃO (PSOL, 3727 votos)
- 7 - ZEZITO MAIO (PMDB, 3590 votos)
- 10 - LUIZ EMANUEL (PPS, 3138 votos) - Coligação: PPS / PROS
- 13 - ROGERINHO PINHEIRO (PHS, 2961 votos) - Coligação: PSB / PHS
- 14 - DEVANIR FERREIRA (PRB, 2948 votos) - Coligação: PRB / PEN / PPL / PSDC / PC do B
- 15 - MAURICIO LEITE (PMDB, 2864 votos)

Eleitos, que se beneficiaram do sistema proporcional:
(com sua posição no ranking de mais votados)

- 17 - MAX DA MATA (PDT, 2339 votos)
- 18 - WANDERSON MARINHO (PSC, 2208 votos) - Coligação: PSC / PSDB
- 20 - SANDRO PARRINI (PDT, 2024 votos)
- 26 - LUIZ PAULO AMORIM (PV, 1696 votos) - Coligação: PP / PV / REDE
- 30 - ROBERTO MARTINS (PTB, 1547 votos)
- 33 - CLEBER FELIX (CLEBINHO) (PP, 1524 votos) - Coligação: PP / PV / REDE

Votação (nominal) das coligações e número de candidatos eleitos:

- 1 - Coligação: PPS / PROS, 32411 votos, 4 candidatos eleitos
- 2 - Coligação: PSB / PHS, 17676 votos, 2 candidatos eleitos
- 3 - Coligação: PSC / PSDB, 16459 votos, 2 candidatos eleitos
- 4 - Coligação: PDT, 16050 votos, 2 candidatos eleitos
- 5 - Coligação: PTB, 15729 votos, 2 candidatos eleitos
- 6 - Coligação: PP / PV / REDE, 14755 votos, 2 candidatos eleitos
- 7 - Coligação: PRB / PEN / PPL / PSDC / PC do B, 11042 votos, 0 candidato eleito
- 8 - Coligação: SD / PMB / PSD / PRTB / PTC, 10572 votos, 1 candidato eleito
- 9 - Coligação: PTN / DEM / PR / PT do B, 10318 votos, 0 candidato eleito
- 10 - Coligação: PMDB, 9119 votos, 0 candidato eleito
- 11 - Coligação: PT, 7885 votos, 0 candidato eleito
- 12 - Coligação: PRP / PMN / PSL, 6178 votos, 0 candidato eleito
- 13 - Coligação: PSOL, 3727 votos, 0 candidato eleito

Votação (nominal) dos partidos e número de candidatos eleitos:

- 1 - PPS, 31036 votos, 4 candidatos eleitos
- 2 - PDT, 16050 votos, 2 candidatos eleitos
- 3 - PTB, 15729 votos, 2 candidatos eleitos
- 4 - PSB, 12029 votos, 2 candidatos eleitos
- 5 - PMDB, 9119 votos, 0 candidato eleito
- 6 - PSDB, 9006 votos, 1 candidato eleito
- 7 - PT, 7885 votos, 0 candidato eleito
- 8 - PV, 7748 votos, 1 candidato eleito
- 9 - PSC, 7453 votos, 1 candidato eleito
- 10 - PP, 6396 votos, 1 candidato eleito
- 11 - PHS, 5647 votos, 0 candidato eleito
- 12 - PRB, 4618 votos, 0 candidato eleito
- 13 - DEM, 4367 votos, 0 candidato eleito
- 14 - PSD, 3899 votos, 1 candidato eleito
- 15 - PSOL, 3727 votos, 0 candidato eleito

```
16 - PTN, 3692 votos, 0 candidato eleito
17 - SD, 3339 votos, 0 candidato eleito
18 - PRP, 3199 votos, 0 candidato eleito
19 - PMB, 2652 votos, 0 candidato eleito
20 - PC do B, 2025 votos, 0 candidato eleito
21 - PSDC, 2020 votos, 0 candidato eleito
22 - PMN, 1762 votos, 0 candidato eleito
23 - PT do B, 1613 votos, 0 candidato eleito
24 - PROS, 1375 votos, 0 candidato eleito
25 - PPL, 1302 votos, 0 candidato eleito
26 - PSL, 1217 votos, 0 candidato eleito
27 - PEN, 1077 votos, 0 candidato eleito
28 - PTC, 682 votos, 0 candidato eleito
29 - PR, 646 votos, 0 candidato eleito
30 - REDE, 611 votos, 0 candidato eleito
```

Total de votos nominais: 171921

3 Condições de Entrega (Java)

Crie um arquivo `vereadores.jar` com as classes compiladas e com um MANIFEST que defina a classe principal do seu programa. O método `main` desta classe deve receber como parâmetro o nome de arquivo CSV com os dados referentes à votação de vereadores em um município, o tipo de encoding usado na produção do arquivo (que varia de acordo com o sistema operacional no qual o Divulga foi executado) e gerar a saída na saída padrão.

Exemplos de comando de execução se o arquivo tiver sido gerado pelo Divulga no Windows:

```
java -jar vereadores.jar vitoria2016.csv ISO8859_1
```

Exemplos de comando de execução se o arquivo tiver sido gerado pelo Divulga no Mac OS ou Linux:

```
java -jar vereadores.jar vitoria2016.csv UTF8
```

O seu programa deve funcionar tanto com a codificação ISO8859_1 quanto com UTF8. A saída deve seguir a mesma codificação da entrada.

Inclua os arquivos binários, o código fonte, e o arquivo de *build* (ver abaixo) e o relatório (PDF) um arquivo .zip nomeado com `vereadores` seguido dos nomes dos componentes do grupo separados por hífen. (Exemplo: `vereadores-JoseSilva-AntonioVieira.zip`). Renomeie a extensão do arquivo .zip para .dat (caso contrário, o seu e-mail será rejeitado pelo sistema de e-mail, Exemplo: `vereadores-JoseSilva-AntonioVieira.dat`). Envie o arquivo para jpalmeyda@inf.ufes.br com o nome do arquivo no campo “subject”.

O relatório deve descrever a sua implementação. Discuta os testes executados (e inclua os arquivos de entrada usados para teste). A completude dos testes é um importante critério de avaliação. Declare bugs conhecidos (“*known bugs*”) no seu relatório. (Bugs conhecidos documentados são melhores do que bugs desconhecidos e não documentados.) Assim como com relação à completude dos testes a qualidade do seu relatório é um importante critério de avaliação.

Arquivo de Build.

Para que o *build* seja feito de forma automatizada, o arquivo de *build* do Ant deve, obrigatoriamente, encontrar-se na raiz da pasta criada e chamar-se *build.xml*. Além disso, ele deve ser feito de forma a responder aos seguintes comandos:

Comando	Resultado esperado
ant compile	O código-fonte deve ser compilado, gerando os arquivos .class para todas as classes do trabalho.
ant jar	Os arquivos .class são empacotados em um arquivo .jar
ant run	O programa deve ser executado especificando o arquivo <i>divulga.csv</i> como parâmetro.
ant clean	Todos os arquivos gerados (classes compiladas) e eventuais arquivos de entrada de dados devem ser excluídos, sobrando somente o conteúdo original do arquivo compactado (ou seja, o código-fonte e o arquivo de <i>build</i>).

Segue abaixo um exemplo de arquivo *build.xml* que atende às especificações acima. Em negrito encontram-se marcados os dados que devem ser adaptados dependendo do projeto:

- **src**: subpasta onde encontra-se todo o código-fonte;
- **bin**: subpasta onde serão colocadas as classes compiladas;
- **seupacote.SuaClassePrincipal**: nome da classe principal do programa, ou seja, aquela que possui o método *main*.

```
<project name="TrabalhoPROG3_2018_2" default="compile" basedir=".">
  <description>Arquivo de build do trabalho de PROG3, 2018/2.</description>

  <!-- Propriedades do build. -->
  <property name="src" location="src" />
  <property name="bin" location="bin" />
  <property name="mainClass" value="seupacote.SuaClassePrincipal" />

  <!-- Inicialização. -->
  <target name="init" description="Inicializa as estruturas necessárias.">
    <tstamp/>
    <mkdir dir="${bin}" />
  </target>

  <!-- Compilação. -->
  <target name="compile" depends="init" description="Compila o código-fonte.">
    <javac includeantruntime="false" srcdir="${src}" destdir="${bin}" />
  </target>

  <!-- Pacote. -->
  <target name="jar" depends="compile">
    <jar destfile="vereadores.jar" basedir="${bin}">
      <manifest>
        <attribute name="Main-Class" value="${mainClass}" />
      </manifest>
    </jar>
  </target>

  <!-- Execução normal. -->
  <target name="run" depends="compile" description="Executa o programa principal, em modo normal.">
    <java classname="${mainClass}">
      <arg value="divulga.csv" />
      <arg value="UTF8" />
      <classpath>
        <pathelement path="${bin}" />
      </classpath>
    </java>
  </target>

  <!-- Limpeza. -->
  <target name="clean" description="Limpa o projeto, deixando apenas o código-fonte." >
    <delete dir="${bin}" />
    <delete><fileset dir="." includes="*.csv" /></delete>
  </target>
</project>
```