UNIVERSIDADE CRUZEIRO DO SUL

Michael Douglas RGM: 1908513-3

Erik Coelho Gama RGM: 1947384-2

Carlos Ivan Uemura Rodrigues RGM: 1947485-7

Igor George Alves RGM: 1948524-7

Técnicas de Desenvolvimento de Algoritmos

Professor: Ivan Carlos

**ÍNDICE**

Página 3 ..............................................................................................Método 1

Página 4 ..............................................................................................Método 2

Página 6 ..............................................................................................Método 3

Página 8 ..............................................................................................Método 4

Página 10.............................................................................................Método 5

Página 13.............................................................................................Método 6

Página 14.............................................................................................Método 7

Página 16.............................................................................................Método 8

Página 18.............................................................................................Método 9

Página 20.............................................................................................Método 10

Página 22.............................................................................................Método 11

Página 26.............................................................................................Método 12

Página 31.............................................................................................Método 13

Página 32.............................................................................................Método 14

**Método 1 – Fornecer o Valor da Loteca:**

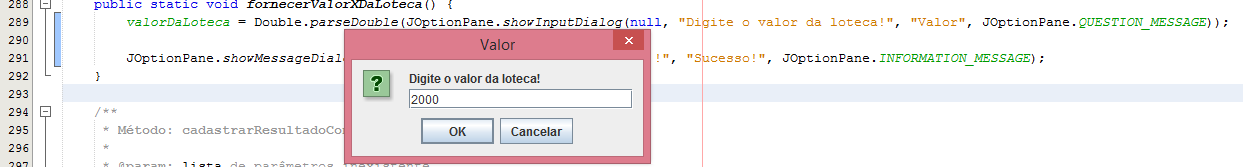
public static void fornecerValorXDaLoteca() {

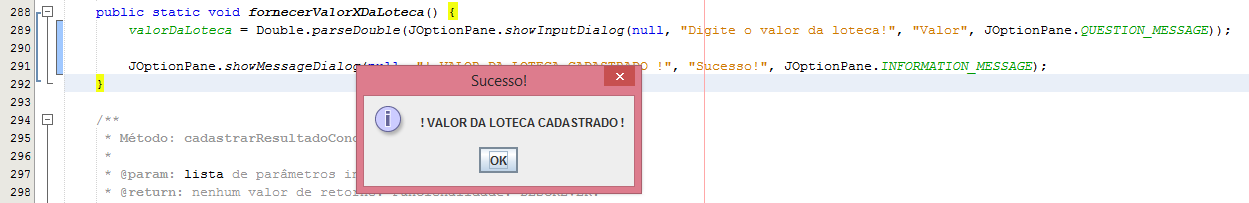
//Cadastra o valor da loteca, guardando na variavel valorDaLoteca

valorDaLoteca = Double.parseDouble(JOptionPane.showInputDialog(null, "Digite o valor da loteca!", "Valor Lotece", JOptionPane.QUESTION\_MESSAGE));

JOptionPane.showMessageDialog(null, "! VALOR DA LOTECA CADASTRADO !", "Cadastro", JOptionPane.INFORMATION\_MESSAGE);

}





**Método 2 - Cadastrar Resultados do Concurso:**

public static void cadastrarResultadoConcurso() {

//Variável criada para ser reutilizada no metodo limpar valores.

resultadosCadastrado = true;

//Cadastra os resultados dos jogos.

for (int i = 0; i < resultadoConcurso.length; i++) {

int jogo = i + 1;

int valorJogo = Integer.parseInt((JOptionPane.showInputDialog("Digite o resultado do Jogo " + jogo

+ "\n\n 1-Para Time Mandante \n 0- Para Empate\n 2-Para Time Visitante")));

//Verificação que obrigada o jogador digitar apenas resultados válidos.

if (valorJogo != 1 && valorJogo != 0 && valorJogo != 2) {

JOptionPane.showMessageDialog(null, "Válidos apenas os números 1 - 0 - 2\n TENTE NOVAMENTE!", "VALOR ERRADO", JOptionPane.INFORMATION\_MESSAGE);

i = i - 1;

} else {

resultadoConcurso[i] = valorJogo;

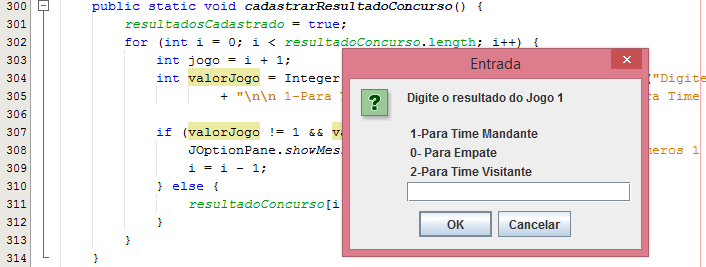
}

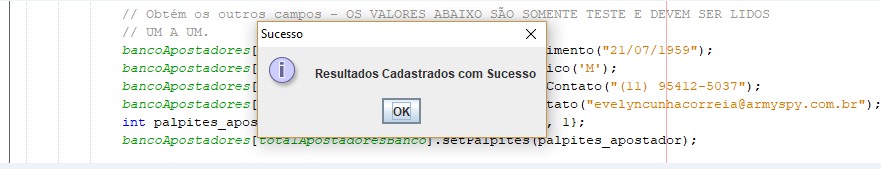
}

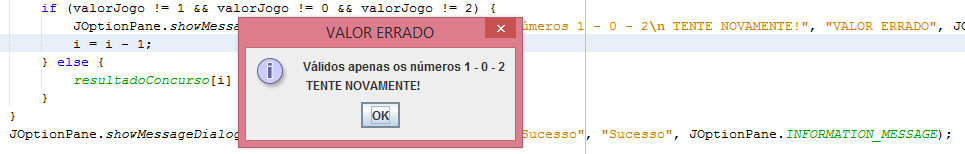
JOptionPane.showMessageDialog(null, "Resultados Cadastrados com Sucesso", "Sucesso", JOptionPane.INFORMATION\_MESSAGE);

}

**Método 2 - Cadastrar Resultados do Concurso: (PRINTS)**







**Método 3 - Cadastrar Apostas de Jogos:**

public static void cadastrarApostasDeJogos() {

/\*Instância da class apostador, o método utiliza os gatters e setters

disponibilizado pelo profressor para dar valor os atributos do Apostador.

\*/

bancoApostadores[totalApostadoresBanco] = new Apostador();

bancoApostadores[totalApostadoresBanco].setNome(JOptionPane.showInputDialog(null, "Digite o Nome do Apostador", "Cadastro do Apostador", JOptionPane.QUESTION\_MESSAGE));

bancoApostadores[totalApostadoresBanco].setDataDeNascimento((JOptionPane.showInputDialog(null, "Digite a Data de Nascimento do Apostador no formato ##/##/##", "Cadastro do Apostador", JOptionPane.QUESTION\_MESSAGE)));

bancoApostadores[totalApostadoresBanco].setSexoBiologico(JOptionPane.showInputDialog(null, "Digite sexo do Apostador H-Homem M-Mulher", "Cadastro do Apostador", JOptionPane.QUESTION\_MESSAGE).charAt(0));

bancoApostadores[totalApostadoresBanco].setEmailDeContato(JOptionPane.showInputDialog(null, "Digite o Email do Apostador", "Cadastro do Apostador", JOptionPane.QUESTION\_MESSAGE));

bancoApostadores[totalApostadoresBanco].setTelefoneDeContato(JOptionPane.showInputDialog(null, "Digite o Telefone do Apostador ####-####", "Cadastro do Apostador", JOptionPane.QUESTION\_MESSAGE));

int palpitesApostador[] = new int[10];

/\*Disponibiliza para o usuário o cadastramento dos palpites do jogador.

\*/

for (int i = 0; i < palpitesApostador.length; i++) {

int jogo = i + 1;

int valorJogo = Integer.parseInt((JOptionPane.showInputDialog("Digite o resultado do Jogo " + jogo

+ "\n\n 1-Para Time Mandante \n 0- Para Empate\n 2-Para Time Visitante")));

//Obriga o usuário a digitar o valor correto, apenas 1 , 0 ou 2;

if (valorJogo != 1 && valorJogo != 0 && valorJogo != 2) {

JOptionPane.showMessageDialog(null, "Válidos apenas os números 1 - 0 - 2\n TENTE NOVAMENTE!", "VALOR ERRADO", JOptionPane.INFORMATION\_MESSAGE);

i = i - 1;

} else {

palpitesApostador[i] = valorJogo;

}

}

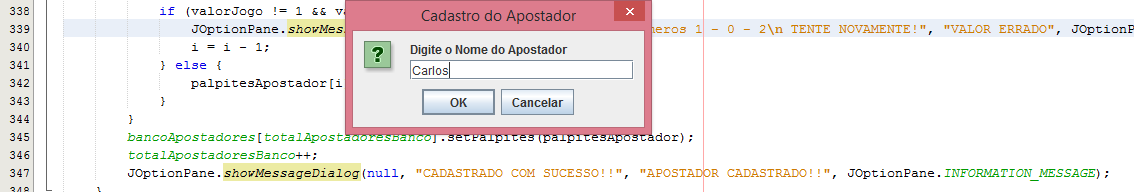
//Seta os palpites no vetor banco.

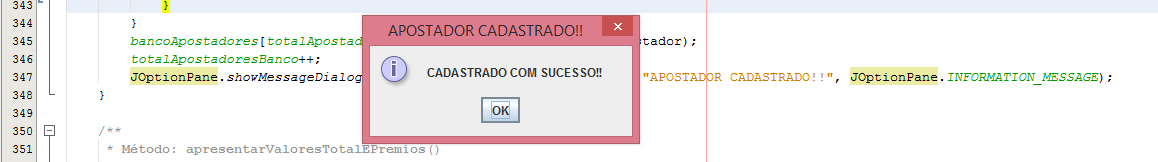
bancoApostadores[totalApostadoresBanco].setPalpites(palpitesApostador);

totalApostadoresBanco++;

JOptionPane.showMessageDialog(null, "CADASTRADO COM SUCESSO!!", "APOSTADOR CADASTRADO!!", JOptionPane.INFORMATION\_MESSAGE);

}





**Método 4 - Apresentar Apostadores:**

public static void apresentarApostadores() {

int cont;

for (cont = 0; cont < totalApostadoresBanco; cont++) {

JOptionPane.showMessageDialog(null, (cont + 1) + "o Apostador\n\n"

+ "nome\_\_\_\_\_\_\_\_\_\_\_\_\_\_: " + bancoApostadores[cont].getNome() + "\n"

+ "data de nascimento\_\_\_: " + bancoApostadores[cont].getDataDeNascimento() + "\n"

+ "sexo biológico\_\_\_\_\_\_\_: " + bancoApostadores[cont].getSexoBiologico() + "\n"

+ "telefone de contato\_\_\_: " + bancoApostadores[cont].getTelefoneDeContato() + "\n"

+ "e-mail de contato\_\_\_\_\_: " + bancoApostadores[cont].getEmailDeContato() + "\n"

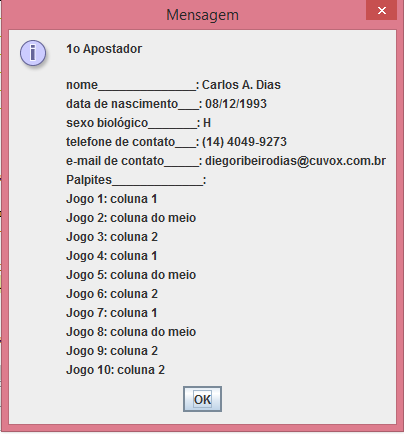
+ "Palpites\_\_\_\_\_\_\_\_\_\_\_\_\_: \n" + msgPalpites(bancoApostadores[cont].getPalpites())

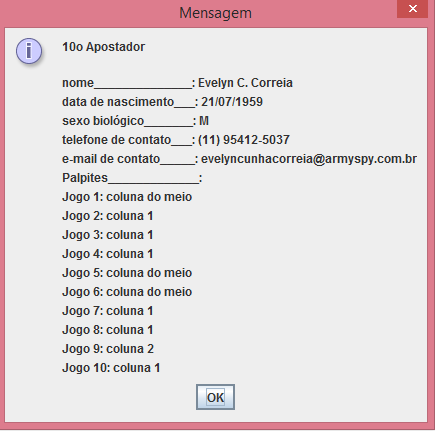
);

}

}

**Método 4 - Apresentar Apostadores: (PRINTS)**





**Método 5 -Apresentar Valores (Total e Prêmios):**

public static void apresentarValoresTotalEPremios() {

JOptionPane.showMessageDialog(null, "Valor da Loteca: " + valorDaLoteca, "", JOptionPane.INFORMATION\_MESSAGE);

//Divide o valor do concurso loteca de acordo com regra do professor.

double quarenta = valorDaLoteca \* 0.4;

double setenta = quarenta \* 0.7;

double trinta = quarenta \* 0.3;

//Pega os acertos dos jogadores, guardando no vetor "palpites";

for (int i = 0; i < totalApostadoresBanco; i++) {

int count = 0;

int palpites[] = bancoApostadores[i].getPalpites();

for (int j = 0; j < 10; j++) {

if (resultadoConcurso[j] == palpites[j]) {

count++;

}

}

totalDeAcertoPlayer[i] = count;

}

int count1 = 0;

int count2 = 0;

/\*Armazena no count1 os jogadores que tiveram 9 acertos,

e armazena no count2 os jogadoresq que tiveram 10 acertos.\*/

for (int i = 0; i < totalApostadoresBanco; i++) {

if (totalDeAcertoPlayer[i] == 9) {

count1++;

} else if (totalDeAcertoPlayer[i] == 10) {

count2++;

}

}

//Informa ao usuário se nenhum jogador acertar 9.

if (count1 == 0) {

JOptionPane.showMessageDialog(null, "Nenhum Jogador Obteve os 9 Acertos!", "Nínguem Acertou 9!", JOptionPane.QUESTION\_MESSAGE);

}

//Informa ao usuário se nenhum jogador acertar 10.

if (count2 == 0) {

JOptionPane.showMessageDialog(null, "Nenhum Jogador Obteve os 10 Acertos!", "Nínguem Acertou 10!", JOptionPane.QUESTION\_MESSAGE);

}

//Informa ao usuário se nenhum jogador acertar 10 e 9.

if (count1 == 0 && count2 == 0) {

JOptionPane.showMessageDialog(null, "Nenhum Jogador Obteve os 10 ou 9 Acertos!", "Nínguem Acertou 10 ou 9!", JOptionPane.QUESTION\_MESSAGE);

} else {

/\*Verifica se o jogador teve 9 ou 10 acertos,

caso ele tenha 9 ou 10, ele será apresentado para o usuário. \*/

for (int i = 0; i < totalApostadoresBanco; i++) {

if (totalDeAcertoPlayer[i] == 9) {

JOptionPane.showMessageDialog(null, "!! Teve 9 Acertos !! \nJogador: " + bancoApostadores[i].getNome() + "\nAcertou: " + totalDeAcertoPlayer[i] + "\nPrêmio: " + trinta / count1, "9 ACERTOS", JOptionPane.QUESTION\_MESSAGE);

} else if (totalDeAcertoPlayer[i] == 10) {

JOptionPane.showMessageDialog(null, "!! Teve 10 Acertos !! \nJogador: " + bancoApostadores[i].getNome() + "\nAcertou: " + totalDeAcertoPlayer[i] + "\nPrêmio: " + setenta / count2, "10 ACERTOS", JOptionPane.QUESTION\_MESSAGE);

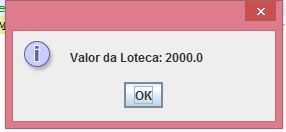
}

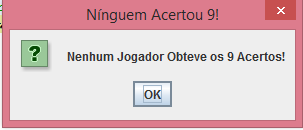
}

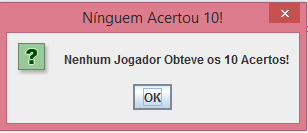
}

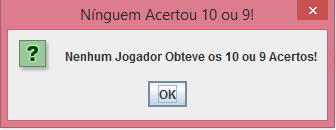
}

**Método 5 -Apresentar Valores (Total e Prêmios): (PRINTS)**









**Método 6 - Apresentar Acertos:**

public static void apresentarAcertos() {

//Guarda no vetor palpites as apostas dos jogadores.

for (int i = 0; i < totalApostadoresBanco; i++) {

int count = 0;

int palpites[] = bancoApostadores[i].getPalpites();

for (int j = 0; j < 10; j++) {

if (resultadoConcurso[j] == palpites[j]) {

count++;

}

}

totalDeAcertoPlayer[i] = count;

}

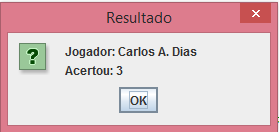
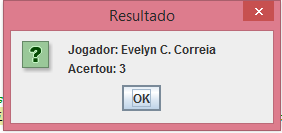
// Mostra quantos acertos os apostadores tiveram.

for (int i = 0; i < totalApostadoresBanco; i++) {

JOptionPane.showMessageDialog(null, "Jogador: " + bancoApostadores[i].getNome() + "\nAcertou: " + totalDeAcertoPlayer[i], "Resultado", JOptionPane.QUESTION\_MESSAGE);

}

}

**Método 7 - Apresentar Jogos Zerados:**

public static void apresentarJogosZerados() {

//Pega a quantidade de acertos de cada apostador.

for (int i = 0; i < totalApostadoresBanco; i++) {

int count = 0;

int palpites[] = bancoApostadores[i].getPalpites();

for (int j = 0; j < 10; j++) {

if (resultadoConcurso[j] == palpites[j]) {

count++;

}

}

totalDeAcertoPlayer[i] = count;

}

int count = 0;

//Apresenta os jogadores que zeraram.

for (int i = 0; i < totalApostadoresBanco; i++) {

if (totalDeAcertoPlayer[i] == 0) {

JOptionPane.showMessageDialog(null, "!!ZEROU !! \nJogador: " + bancoApostadores[i].getNome() + "\nAcertou: " + totalDeAcertoPlayer[i], "0 ACERTOS", JOptionPane.QUESTION\_MESSAGE);

count++;

}

}

//Apresenta uma caixa de texto, se nenhum jogador zerar.

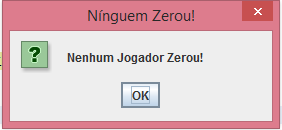
if (count == 0) {

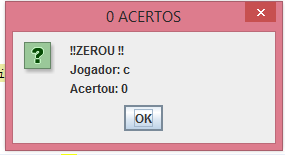
JOptionPane.showMessageDialog(null, "Nenhum Jogador Zerou!", "Nínguem Zerou!", JOptionPane.QUESTION\_MESSAGE);

}

}

**Método 7 - Apresentar Jogos Zerados: (PRINTS)**





**Método 8 - Apresentar Ganhadores 10 Acertos.**

public static void apresentarGanhadores10Acertos() {

//Pega a quantidade de acertos de cada apostador.

for (int i = 0; i < totalApostadoresBanco; i++) {

int count = 0;

int palpites[] = bancoApostadores[i].getPalpites();

for (int j = 0; j < 10; j++) {

if (resultadoConcurso[j] == palpites[j]) {

count++;

}

}

totalDeAcertoPlayer[i] = count;

}

int count = 0;

/\* Pecorre o vetor Total totalApostadoresBanco e compara os resultados

utilizando o vetor totalDeAcertoPlayer, ambos utilziando a variavel/contador i

e verifica se o apostador teve 9 acertos.

\*/

for (int i = 0; i < totalApostadoresBanco; i++) {

if (totalDeAcertoPlayer[i] == 10) {

JOptionPane.showMessageDialog(null, "!! ACERTOU TUDO !! \nJogador: " + bancoApostadores[i].getNome() + "\nAcertou: " + totalDeAcertoPlayer[i], "10 ACERTOS", JOptionPane.QUESTION\_MESSAGE);

count++;

}

}

//Verifica se todos o jogadores não acertaram 10;

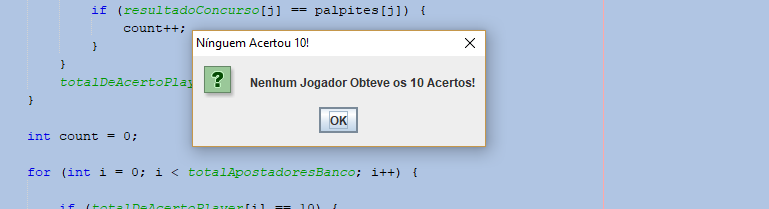
if (count == 0) {

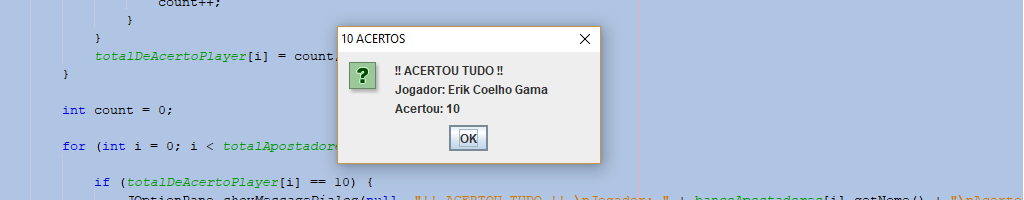
JOptionPane.showMessageDialog(null, "Nenhum Jogador Obteve os 10 Acertos!", "Nínguem Acertou 10!", JOptionPane.QUESTION\_MESSAGE);

}

}

**Prints do Método 8 - Apresentar Ganhadores 10 Acertos.**





**Método 9 - Apresentar Ganhadores 9 Acertos.**

public static void apresentarGanhadores9Acertos() {

//Pega a quantidade de acertos de cada apostador.

for (int i = 0; i < totalApostadoresBanco; i++) {

int count = 0;

int palpites[] = bancoApostadores[i].getPalpites();

for (int j = 0; j < 10; j++) {

if (resultadoConcurso[j] == palpites[j]) {

count++;

}

}

totalDeAcertoPlayer[i] = count;

}

int count = 0;

//Verifica se o apostador teve 9 acertos.

for (int i = 0; i < totalApostadoresBanco; i++) {

if (totalDeAcertoPlayer[i] == 9) {

JOptionPane.showMessageDialog(null, "!! ACERTOU 9 !! \nJogador: " + bancoApostadores[i].getNome() + "\nAcertou: " + totalDeAcertoPlayer[i], "9 ACERTOS", JOptionPane.QUESTION\_MESSAGE);

count++;

}

}

//Verifica se todos o jogadores não acertaram 9;

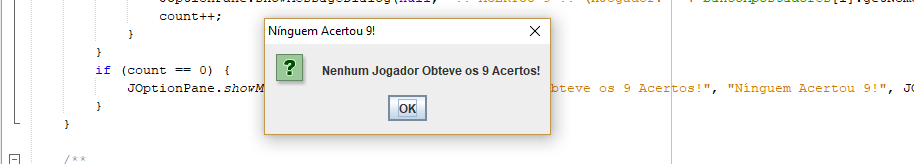
if (count == 0) {

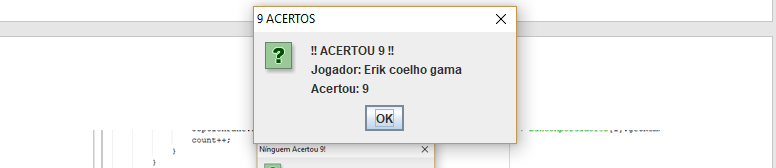
JOptionPane.showMessageDialog(null, "Nenhum Jogador Obteve os 9 Acertos!", "Nínguem Acertou 9!", JOptionPane.QUESTION\_MESSAGE);

}

}

**Prints do Método Apresentar Ganhadores 9 Acertos.**





**Método 10 - Percentual Sexo Biológico.**

public static void apresentarPercentualSexoBiologico() {

int countM = 0;

int countF = 0;

float totalPessoas = 0;

//Conta a quantidade de pessoas Masculina e Feminina, armazenando os homens no countM e as mulheres no countF.

for (int i = 0; i < totalApostadoresBanco; i++) {

String sexo = String.valueOf(bancoApostadores[i].getSexoBiologico());

if (sexo.equals("H") || sexo.equals("h")) {

countM++;

} else if (sexo.equals("M") || sexo.equals("m")) {

countF++;

}

}

//Calcula a quantidade de homens e mulheres, e apresenta para o usuário.

totalPessoas = countM + countF;

String resultH = Float.toString(((countM \* 100) / totalPessoas));

String resultM = Float.toString(((countF \* 100) / totalPessoas));

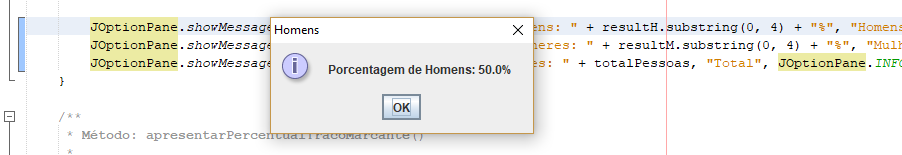
JOptionPane.showMessageDialog(null, "Porcentagem de Homens: " + resultH.substring(0, 4) + "%", "Homens", JOptionPane.INFORMATION\_MESSAGE);

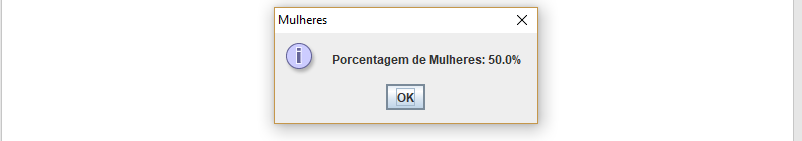
JOptionPane.showMessageDialog(null, "Porcentagem de Mulheres: " + resultM.substring(0, 4) + "%", "Mulheres", JOptionPane.INFORMATION\_MESSAGE);

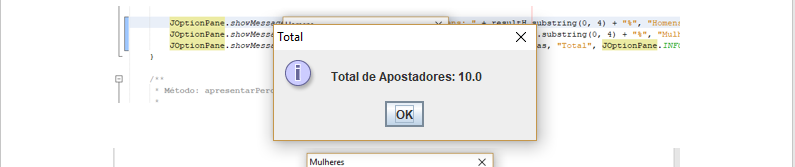
JOptionPane.showMessageDialog(null, "Total de Apostadores: " + totalPessoas, "Total", JOptionPane.INFORMATION\_MESSAGE);

}

**Prints do Método 10 - Percentual Sexo Biológico.**







**Método 11 - Apresentar Percentual Traço Marcante.**

public static void apresentarPercentualTracoMarcante() {

//Matriz interna de traço marcante

double[][] traco = new double[9][2];

String msg = "";

//Preencher Primeira coluna com os códigos dos traços marcantes.

for (int i = 0; i < 9; i++) {

traco[i][0] = i;

}

/\* Adiciona quantidade de pessoas com detemirnado traço na segunda coluna da matriz "traco",

utiliza o metodo, 'tracoMarcantePersonalidade'\*/

for (int cont = 0; cont < totalApostadoresBanco; cont++) {

switch (bancoApostadores[cont].tracoMarcantePersonalidade()) {

case 0:

traco[0][1]++;

break;

case 1:

traco[1][1]++;

break;

case 2:

traco[2][1]++;

break;

case 3:

traco[3][1]++;

break;

case 4:

traco[4][1]++;

break;

case 5:

traco[5][1]++;

break;

case 6:

traco[6][1]++;

break;

case 7:

traco[7][1]++;

break;

case 8:

traco[8][1]++;

break;

}

}

//Ordena a matriz em ordem crescente.

for (int k = 0; k < traco.length; k++) {

for (int j = k; j < traco.length; j++) {

if (traco[k][1] > traco[j][1]) {

double[] vetorTemporario = traco[k];

traco[k] = traco[j];

traco[j] = vetorTemporario;

}

}

}

//Armazena a menssagem com os traços marcantes.

for (int x = 0; x < 9; x++) {

if (traco[x][0] == 0) {

msg = msg + "Irresistível: " + traco[x][1] + "%\n";

} else if (traco[x][0] == 1) {

msg = msg + "Impetuoso: " + traco[x][1] + "%\n";

} else if (traco[x][0] == 2) {

msg = msg + "Discreto: " + traco[x][1] + "%\n";

} else if (traco[x][0] == 3) {

msg = msg + "Amoroso: " + traco[x][1] + "%\n";

} else if (traco[x][0] == 4) {

msg = msg + "Tímido: " + traco[x][1] + "%\n";

} else if (traco[x][0] == 5) {

msg = msg + "Paquerador: " + traco[x][1] + "%\n";

} else if (traco[x][0] == 6) {

msg = msg + "Estudioso: " + traco[x][1] + "%\n";

} else if (traco[x][0] == 7) {

msg = msg + "Sonhador: " + traco[x][1] + "%\n";

} else if (traco[x][0] == 8) {

msg = msg + "Charmoso: " + traco[x][1] + "%\n";

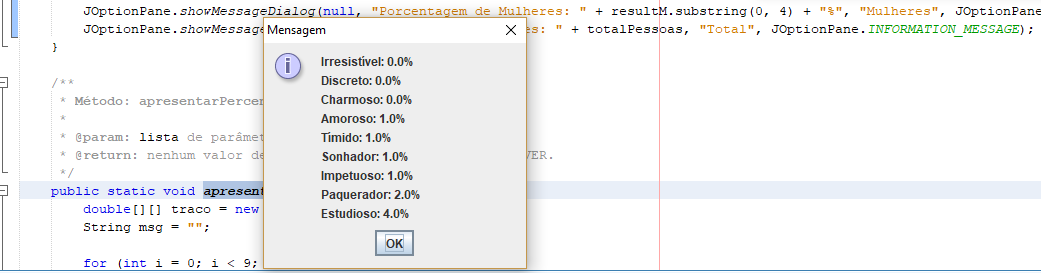
}

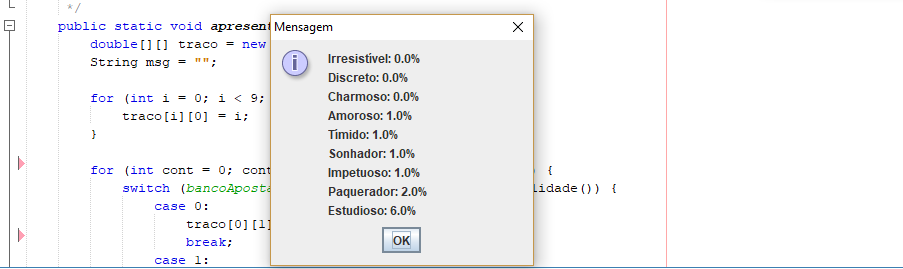
}

JOptionPane.showMessageDialog(null, msg);

}

**Prints do Método - 11 Apresentar Percentual Traço Marcante.**





**Método 12 – Apresentar Traço Ganhadores 10Acertos.**

public static void apresentarTracoGanhadores10Acertos() {

for (int i = 0; i < totalApostadoresBanco; i++) {

int count = 0;

int palpites[] = bancoApostadores[i].getPalpites();

for (int j = 0; j < 10; j++) {

if (resultadoConcurso[j] == palpites[j]) {

count++;

}

}

totalDeAcertoPlayer[i] = count;

}

int count = 0;

int soma;

int ganhadores10Acertos = 0;

for (int i = 0; i < totalApostadoresBanco; i++) {

if (totalDeAcertoPlayer[i] == 10) {

ganhadores10Acertos++;

}

}

if (ganhadores10Acertos != 0) {

double[][] traco = new double[9][2];

String msg = "";

for (int i = 0; i < 9; i++) {

traco[i][0] = i;

}

for (int cont = 0; cont < ganhadores10Acertos; cont++) {

switch (bancoApostadores[cont].tracoMarcantePersonalidade()) {

case 0:

traco[0][1]++;

break;

case 1:

traco[1][1]++;

break;

case 2:

traco[2][1]++;

break;

case 3:

traco[3][1]++;

break;

case 4:

traco[4][1]++;

break;

case 5:

traco[5][1]++;

break;

case 6:

traco[6][1]++;

break;

case 7:

traco[7][1]++;

break;

case 8:

traco[8][1]++;

break;

}

}

for (int k = 0; k < traco.length; k++) {

for (int j = k; j < traco.length; j++) {

if (traco[k][1] < traco[j][1]) {

double[] vetorTemporario = traco[k];

traco[k] = traco[j];

traco[j] = vetorTemporario;

}

}

}

for (int x = 0; x < 9; x++) {

if (traco[x][0] == 0) {

msg = msg + "Irresistível: " + traco[x][1] + "%\n";

} else if (traco[x][0] == 1) {

msg = msg + "Impetuoso: " + traco[x][1] + "%\n";

} else if (traco[x][0] == 2) {

msg = msg + "Discreto: " + traco[x][1] + "%\n";

} else if (traco[x][0] == 3) {

msg = msg + "Amoroso: " + traco[x][1] + "%\n";

} else if (traco[x][0] == 4) {

msg = msg + "Tímido: " + traco[x][1] + "%\n";

} else if (traco[x][0] == 5) {

msg = msg + "Paquerador: " + traco[x][1] + "%\n";

} else if (traco[x][0] == 6) {

msg = msg + "Estudioso: " + traco[x][1] + "%\n";

} else if (traco[x][0] == 7) {

msg = msg + "Sonhador: " + traco[x][1] + "%\n";

} else if (traco[x][0] == 8) {

msg = msg + "Charmoso: " + traco[x][1] + "%\n";

}

}

JOptionPane.showMessageDialog(null, msg);

}

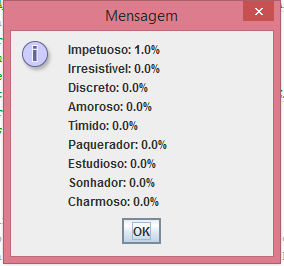
if (ganhadores10Acertos == 0) {

JOptionPane.showMessageDialog(null, "Nenhum Jogador Obteve os 10 Acertos!", "Nínguem Acertou 10!", JOptionPane.QUESTION\_MESSAGE);

}

}

**Prints Método 12 – Apresentar Traço Ganhadores 10Acertos.**



**Método 13 - Limpar Resultados Do Concurso.**

public static void limparResultadosDoConcurso() {

//Verifica se os resultados do concurso foram cadastrados.

if (!resultadosCadastrado) {

JOptionPane.showMessageDialog(null, "Os Resultados não foram Cadastrados!", "Resultados não Cadastrados!", JOptionPane.INFORMATION\_MESSAGE);

} else {

//Zeram todos os resultados dos jogos.

for (int i = 0; i < resultadoConcurso.length; i++) {

resultadoConcurso[i] = -1;

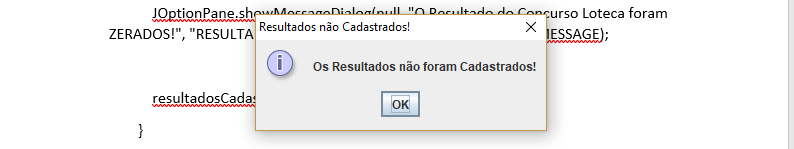
}

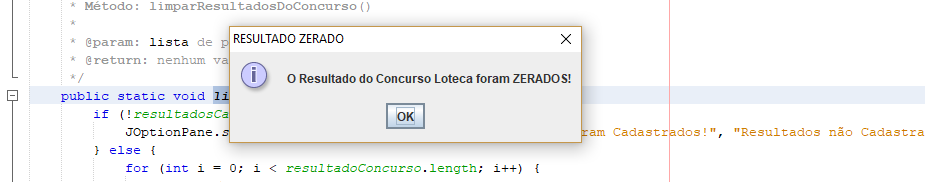
JOptionPane.showMessageDialog(null, "O Resultado do Concurso Loteca foram ZERADOS!", "RESULTADO ZERADO", JOptionPane.INFORMATION\_MESSAGE);

resultadosCadastrado = false;

}

}





**Método 14 - Verificar Termino do Programa.**

public static int verificarTerminoDoSistema() {

//Apresenta uma mensagem para o usuario, perguntando se ele deseja finalizar o programa.

int resposta = JOptionPane.showConfirmDialog(null, "Deseja realmente finalizar o sistema?",

"Finalizar Sistema",

JOptionPane.YES\_NO\_OPTION);

// Se resposta fornecida for igual a YES.

if (resposta == JOptionPane.YES\_OPTION) {

return TERMINA\_SISTEMA;

} else {

return NAO\_TERMINA\_SISTEMA;

}

}

