|  |
| --- |
| trabalho.de.pooi |
| package trabalho.de.pooi;  import java.io.FileReader;  import java.io.IOException;  import java.util.ArrayList;  import java.util.List;  import java.util.Scanner;  import java.util.logging.Level;  import java.util.logging.Logger;  public class TrabalhoDePOOI {  private static List getArmasNigeria(String arquivo) {  int quantidade = -1;  Object arma;  int codigo;  int serial;  boolean acessorio;  int bocal;  List armas = null;  try (FileReader arq = new FileReader(arquivo)) {  Scanner linha = new Scanner(arq);  if (linha.hasNext()) {  quantidade = linha.nextInt();  armas = new ArrayList(quantidade);  }  if (quantidade > 0) {  int count = 0;  while (linha.hasNext()) {  codigo = linha.nextInt();  serial = linha.nextInt();  arma = null;  switch (codigo) {  case 1:  arma = new Estilingue\_de\_Suri(serial, 1);  break;  case 2:  acessorio = linha.nextBoolean();  arma = new Pistola\_Canked(serial, 5, 40, acessorio);  break;  case 3:  arma = new Metralhadora\_Ninek(serial, 20, 80);  break;  case 4:  arma = new Fuzil\_K777(serial, 50, 100);  break;  case 5:  bocal = linha.nextInt();  arma = new Bazuka\_Zonka(serial, 200, 1000, bocal);  break;  default:  break;  }  armas.add(arma);  count++;  }  }  arq.close();  } catch (IOException ex) {  Logger.getLogger(Estoque.class.getName()).log(Level.SEVERE, null, ex);  }  return armas;  }  private static List getArmasBokoHaram(String arquivo) {  int quantidade = 0;  Object arma;  int codigo;  int serial;  boolean acessorio;  int bocal = -1;  List armas = null;  try (FileReader arq = new FileReader(arquivo)) {  Scanner linha = new Scanner(arq);  if (linha.hasNext()) {  quantidade = linha.nextInt();  armas = new ArrayList(quantidade);  }  if (quantidade > 0) {  int count = 0;  while (linha.hasNext()) {  codigo = linha.nextInt();  serial = linha.nextInt();  arma = null;  switch (codigo) {  case 1:  acessorio = linha.nextBoolean();  arma = new Zarabatana\_Longin(serial, 2, acessorio);  break;  case 2:  acessorio = linha.nextBoolean();  arma = new Lanca\_Foral(serial, 1, acessorio);  break;  case 3:  arma = new Metralhadora\_HamHam(serial, 10, 80);  break;  case 4:  arma = new Lanca\_Chamas\_Mirak(serial, 40, 50);  break;  case 5:  acessorio = linha.nextBoolean();  arma = new Lanca\_Foguetes\_OnOn(serial, 150, 1000, acessorio);  break;  default:  break;  }  armas.add(arma);  count++;  }  }  arq.close();  } catch (IOException ex) {  Logger.getLogger(Estoque.class.getName()).log(Level.SEVERE, null, ex);  }  return armas;  }  private static int armasArtesanaisNigeria(List armas) {  int total = 0;  Object arma;  String nomeClasse;  for (int i = 0; i < armas.size(); i++) {  arma = armas.get(i);  nomeClasse = arma.getClass().getSimpleName();  switch (nomeClasse) {  case "Estilingue\_de\_Suri":  total++;  break;  default:  break;  }  }  if (total > 0) {  System.out.println("O exército nigeriano produz " + total + " arma artesanal por dia.");  }  return total;  }  private static int armasArtesanaisBokoHaram(List armas) {  int total = 0;  Object arma;  String nomeClasse;  for (int i = 0; i < armas.size(); i++) {  arma = armas.get(i);  nomeClasse = arma.getClass().getSimpleName();  switch (nomeClasse) {  case "Zarabatana\_Longin":  total++;  break;  case "Lanca\_Foral":  total++;  break;  default:  break;  }  }  if (total > 0) {  System.out.println("O Boko Haram produz " + total + " arma artesanal por dia.");  }  return total;  }  private static void bocalBazuka\_Zonka(List armas) {  int maiorBocal = 0;  List<Bazuka\_Zonka> bazuka = new ArrayList(armas.size());  Bazuka\_Zonka arma;  Object atual;  String nomeClasse;  for (int i = 0; i < armas.size(); i++) {  atual = armas.get(i);  nomeClasse = atual.getClass().getSimpleName();  switch (nomeClasse) {  case "Bazuka\_Zonka":  arma = (Bazuka\_Zonka) atual;  if (maiorBocal < arma.getBocal()) {  maiorBocal = arma.getBocal();  if (!(bazuka.isEmpty())) {  bazuka.clear();  }  bazuka.add(arma);  } else if (maiorBocal == arma.getBocal()) {  bazuka.add(arma);  }  break;  default:  break;  }  }  if (!(bazuka.isEmpty())) {  if (bazuka.size() == 1) {  System.out.println("A Bazuka Zonka de serial " + bazuka.get(0).getSerial() + " é a que possui mais bocais (" + bazuka.get(0).getBocal() + " bocais).");  } else {  System.out.println("EMPATE: ");  for (int i = 0; i < bazuka.size() - 1; i++) {  System.out.print(bazuka.get(i).getSerial() + ", ");  }  System.out.print(bazuka.get(bazuka.size() - 1).getSerial() + ".");  }  } else {  System.out.println("Não há Bazuka Zonka.");  }  }  private static void qtdLancaFoguetesEspeciais(List armas) {  List<Lanca\_Foguetes\_OnOn> normais = new ArrayList(armas.size());  List<Lanca\_Foguetes\_OnOn> especiais = new ArrayList(armas.size());  Lanca\_Foguetes\_OnOn arma;  Object atual;  String nomeClasse;  for (int i = 0; i < armas.size(); i++) {  atual = armas.get(i);  nomeClasse = atual.getClass().getSimpleName();  switch (nomeClasse) {  case "Lanca\_Foguetes\_OnOn":  arma = (Lanca\_Foguetes\_OnOn) atual;  if (!(arma.getAcessorio())) {  normais.add(arma);  } else {  especiais.add(arma);  }  break;  default:  break;  }  }  if (normais.size() < especiais.size()) {  System.out.println("SIM.");  } else if (normais.size() > especiais.size()) {  System.out.println("NÃO.");  } else {  System.out.println("EMPATE.");  }  }  private static void calibreMaior100(List nigeria, List boko\_haram) {  ArrayList maior100 = new ArrayList();  Object atual;  String nomeClasse;  for (int i = 0; i < nigeria.size(); i++) {  atual = nigeria.get(i);  nomeClasse = atual.getClass().getGenericSuperclass().getTypeName();  switch (nomeClasse) {  case "trabalho.de.pooi.Profissional":  if (((Profissional) atual).getCalibre() > 100) {  maior100.add(((Profissional) atual).getSerial());  }    break;  default:  break;  }  }  for (int i = 0; i < boko\_haram.size(); i++) {  atual = boko\_haram.get(i);  nomeClasse = atual.getClass().getGenericSuperclass().getTypeName();  switch (nomeClasse) {  case "trabalho.de.pooi.Profissional":  if (((Profissional) atual).getCalibre() > 100) {  maior100.add(((Profissional) atual).getSerial());  }  break;  default:  break;  }  }  System.out.print("As armas com calibre maior que 100 são: ");  for (int i = 0; i < maior100.size() - 1; i++) {  System.out.print(maior100.get(i) + ", ");  }  System.out.println(maior100.get(maior100.size() - 1) + ".");  }    public static int diasEstoque(Estoque estoque, List armas, String nome) {  int pontosTotal = 0, parar = 0, dias = 0;  Object arma;  boolean continuar = true;    while(parar < armas.size()){  parar = 0;  for (int i = 0; (i < armas.size()) && (parar < armas.size()); i++) {    arma = armas.get(i);  continuar = ((Arma) arma).estoqueDec(estoque);    if (continuar == true) {  pontosTotal += ((Arma) arma).getPontuacao();  if(i + 1 == armas.size()){  dias++;  }  } else {  parar++;  }  }  }    System.out.println("O " + nome + " chegou à pontuação de " + pontosTotal + ".");    return dias;  }  public static void main(String[] args) {    String arq\_estoque\_nigeria = "C:\\Users\\2018122760119\\Downloads\\estoque\_exercito\_nigeriano.txt";  String arq\_armas\_nigeria = "C:\\Users\\2018122760119\\Downloads\\armas\_exercito\_nigeriano.txt";  String arq\_estoque\_boko\_haram = "C:\\Users\\2018122760119\\Downloads\\estoque\_boko\_haram.txt";  String arq\_armas\_boko\_haram = "C:\\Users\\2018122760119\\Downloads\\armas\_boko\_haram.txt";    Estoque estoque\_nigeria = new Estoque(arq\_estoque\_nigeria);  Estoque estoque\_boko\_haram = new Estoque(arq\_estoque\_boko\_haram);    List armas\_nigeria = getArmasNigeria(arq\_armas\_nigeria);  List armas\_boko\_haram = getArmasBokoHaram(arq\_armas\_boko\_haram);    int artesanalNigeria = armasArtesanaisNigeria(armas\_nigeria);  int artesanalBokoHaram = armasArtesanaisBokoHaram(armas\_boko\_haram);    if ((artesanalNigeria + artesanalBokoHaram) > 0) {  System.out.println("O total de armas artesanais produzidas é " + (artesanalNigeria + artesanalBokoHaram) + ".");  }    bocalBazuka\_Zonka(armas\_nigeria);  qtdLancaFoguetesEspeciais(armas\_boko\_haram);  calibreMaior100(armas\_nigeria, armas\_boko\_haram);  int bokoDias = diasEstoque(estoque\_boko\_haram, armas\_boko\_haram, "Boko Haram");  int nigerDias = diasEstoque(estoque\_nigeria, armas\_nigeria, "exército nigeriano");  System.out.println("O Boko Haram possui estoque para " + bokoDias + " dias.");  System.out.println("O exército nigeriano possui estoque para " + nigerDias + " dias.");    }  } |

|  |
| --- |
| Arma |
| package trabalho.de.pooi;  public abstract class Arma {    protected int serial;  //private final int codigo;  protected int pontuacao;    protected int madeira = 0;  protected int aco = 0;  protected int ferro = 0;  protected int aluminio = 0;  protected int chumbo = 0;  protected int cobre = 0;  protected int ouro = 0;  protected int prata = 0;  protected int couro = 0;  protected int pedra = 0;    public Arma(int serial, int pontuacao) {    this.serial = serial;  this.pontuacao = pontuacao;    }    private static boolean validarDec(int estoque, int valor) {  return estoque >= valor;  }    public boolean estoqueDec(Estoque estoque){    /\*System.out.println("MADEIRA: " + estoque.getMadeira() + "\tCUSTO: " + this.madeira);  System.out.println("AÇO: " + estoque.getAco() + "\tCUSTO: " + this.aco);  System.out.println("FERRO: " + estoque.getFerro() + "\tCUSTO: " + this.ferro);  System.out.println("ALUMÍNIO: " + estoque.getAluminio() + "\tCUSTO: " + this.aluminio);  System.out.println("CHUMBO: " + estoque.getChumbo() + "\tCUSTO: " + this.chumbo);  System.out.println("COBRE: " + estoque.getCobre() + "\tCUSTO: " + this.cobre);  System.out.println("OURO: " + estoque.getOuro() + "\tCUSTO: " + this.ouro);  System.out.println("PRATA: " + estoque.getPrata() + "\tCUSTO: " + this.prata);  System.out.println("COURO: " + estoque.getCouro() + "\tCUSTO: " + this.couro);  System.out.println("PEDRA: " + estoque.getPedra() + "\tCUSTO: " + this.pedra);\*/    if (validarDec(estoque.getMadeira(), this.madeira) &&  validarDec(estoque.getAco(), this.aco) &&  validarDec(estoque.getFerro(), this.ferro) &&  validarDec(estoque.getAluminio(), this.aluminio) &&  validarDec(estoque.getChumbo(), this.chumbo) &&  validarDec(estoque.getCobre(), this.cobre) &&  validarDec(estoque.getOuro(), this.ouro) &&  validarDec(estoque.getPrata(), this.prata) &&  validarDec(estoque.getCouro(), this.couro) &&  validarDec(estoque.getPedra(), this.pedra)){  //System.out.println("ENTROU AQUI.");  estoque.setMadeira(estoque.getMadeira() - this.madeira);  estoque.setAco(estoque.getAco() - this.aco);  estoque.setFerro(estoque.getFerro() - this.ferro);  estoque.setAluminio(estoque.getAluminio() - this.aluminio);  estoque.setChumbo(estoque.getChumbo() - this.chumbo);  estoque.setCobre(estoque.getCobre() - this.cobre);  estoque.setOuro(estoque.getOuro() - this.ouro);  estoque.setPrata(estoque.getPrata() - this.prata);  estoque.setCouro(estoque.getCouro() - this.couro);  estoque.setPedra(estoque.getPedra() - this.pedra);    return true;    }    return false;    }  public int getSerial() {  return serial;  }  public int getPontuacao() {  return pontuacao;  }    } |
| Artesanal |
| package trabalho.de.pooi;  public abstract class Artesanal extends Arma {  public Artesanal(int serial, int pontuacao) {  super(serial, pontuacao);  }    } |

|  |
| --- |
| Profissional |
| package trabalho.de.pooi;  public abstract class Profissional extends Arma {    private int calibre;  public Profissional(int serial, int pontuacao, int calibre) {  super(serial, pontuacao);  this.calibre = calibre;  }  public int getCalibre() {  return calibre;  }    } |

|  |
| --- |
| Bazuka\_Zonka |
| package trabalho.de.pooi;  public class Bazuka\_Zonka extends Profissional {  private int bocal;  public Bazuka\_Zonka(int serial, int pontuacao, int calibre, int bocal) {  //super(serial, 5, 200, 1000);  super(serial, pontuacao, calibre);    this.bocal = bocal;  this.aco = 1000;  this.ferro = 1000;  this.aluminio = 500;  this.chumbo = 20;  this.prata = 200;  this.pedra = 100;  if (this.bocal > 0) {  this.aco += bocal \* 200;  this.ouro += bocal \* 100;  this.pontuacao += bocal \* 100;  }  }  public int getBocal() {  return bocal;  }  } |

|  |
| --- |
| Fuzil\_K777 |
| package trabalho.de.pooi;  public class Fuzil\_K777 extends Profissional {  public Fuzil\_K777(int serial, int pontuacao, int calibre) {    //super(serial, 4, 50, 100);  super(serial, pontuacao, calibre);    this.ferro = 800;  this.aluminio = 20;  this.chumbo = 20;  this.cobre = 20;  this.ouro = 20;  this.prata = 20;  this.pedra = 50;    }  } |

|  |
| --- |
| Lanca\_Chamas\_Mirak |
| package trabalho.de.pooi;  public class Lanca\_Chamas\_Mirak extends Profissional {  public Lanca\_Chamas\_Mirak(int serial, int pontuacao, int calibre) {    //super(serial, 4, 40, 50);  super(serial, pontuacao, calibre);    this.madeira = 500;  this.aco = 20;  this.ferro = 20;  this.aluminio = 20;  this.chumbo = 20;  this.cobre = 10;  this.ouro = 10;  this.prata = 10;  this.pedra = 20;    }  } |

|  |
| --- |
| Pistola\_Canked |
| package trabalho.de.pooi;  public class Pistola\_Canked extends Profissional {    private boolean acessorio;  public Pistola\_Canked(int serial, int pontuacao, int calibre, boolean acessorio) {    //super(serial, 2, 5, 40);  super(serial, pontuacao, calibre);    this.acessorio = acessorio;  this.madeira = 50;  this.aco = 100;  this.ferro = 50;  this.chumbo = 20;  this.cobre = 10;  this.ouro = 30;  this.prata = 20;    if(this.acessorio){  this.prata = 80;  this.pontuacao = 15;  }    }    } |

|  |
| --- |
| Lanca\_Foguetes\_OnOn |
| package trabalho.de.pooi;  public class Lanca\_Foguetes\_OnOn extends Profissional {    private boolean acessorio;  public Lanca\_Foguetes\_OnOn(int serial, int pontuacao, int calibre, boolean acessorio) {    //super(serial, 5, 150, 1000);  super(serial, pontuacao, calibre);    this.acessorio = acessorio;  this.aco = 1000;  this.ferro = 1000;  this.aluminio = 400;  this.chumbo = 20;  this.ouro = 10;    if(this.acessorio){  this.prata = 500;  this.pontuacao = 300;  }    }  public boolean getAcessorio() {  return acessorio;  }    } |

|  |
| --- |
| Metralhadora\_HamHam |
| package trabalho.de.pooi;  public class Metralhadora\_HamHam extends Profissional {  public Metralhadora\_HamHam(int serial, int pontuacao, int calibre) {    //super(serial, 3, 10, 80);  super(serial, pontuacao, calibre);    this.madeira = 20;  this.aco = 50;  this.ferro = 50;  this.aluminio = 50;  this.chumbo = 50;  this.couro = 20;  this.pedra = 20;    }  } |
| Metralhadora\_Ninek |
| package trabalho.de.pooi;  public class Metralhadora\_Ninek extends Profissional {  public Metralhadora\_Ninek(int serial, int pontuacao, int calibre) {    //super(serial, 3, 20, 80);  super(serial, pontuacao, calibre);    this.madeira = 200;  this.aco = 200;  this.ferro = 200;  this.ouro = 10;  this.prata = 10;    }  } |