**Teste 17 março**

1-F

2-F

3-F

7-V

13-V

14-F

17-F

19-n=0

20-i=3

21-V

22-dá Erro

25-F

26-V

27-bool=false, porque sao objetos diferentes que têm espaço de memoria diferente.

28-F

29-F

30-V

31- Stdout.println(aa4.str);

32-F

33- Stdout.println(bb1.str);

**Teste 5 abril**

**Parte 1**

1-V

2-F

3-V

4-V

5-V

6-F

7-F

8-F

9-V

10-F

11-F

**Parte 2**

1-d)

2-b)

3-b)

4-b)

5-d)

6-a)

7-b)

8-b)

9-b)

**Código Football Teams**

**Interface – Team management**

*package* edu.ufp.inf.lp2.Teste;  
  
*public interface TeamManagmentI* {  
 *public void* hirePlayer(Player p);  
 *public* Player sellPlayer(*int* number);  
 *public* Player transferPlayer(*int* number,*TeamManagmentI* team);  
}

**Class – Team**

*package* edu.ufp.inf.lp2.Teste;  
  
*import* edu.princeton.cs.algs4.StdOut;  
*import* edu.ufp.inf.lp2.intro\_01.Date;  
  
*import* java.util.ArrayList;  
*import* java.util.HashMap;  
*import* java.util.Objects;  
  
*public class* Team *implements TeamManagmentI*{  
 *private* String name;  
 *private* ArrayList<Player> players = *new* ArrayList<>(20);  
 *private* ArrayList<GameCall> games = *new* ArrayList<>();  
 *public* Team(String name,ArrayList<Player> players){  
 *this*.name=name;  
 *this*.players=players;  
 }  
 *public* String getName(){  
 *return this*.name;  
 }  
 *public void* setName(String name){  
 *this*.name=name;  
 }  
 *public* ArrayList<Player> getPlayers(){  
 *return this*.players;  
 }  
  
 *public* ArrayList<GameCall> getGames() {  
 *return* games;  
 }  
  
 *public void* setGames(ArrayList<GameCall> games) {  
 *this*.games = games;  
 }  
  
 *public void* setPlayers(ArrayList<Player> players){  
 *this*.players=players;  
 }  
 *public* GameCall createGameCall(Date d, Team visitingTeam){  
 HashMap<Object,Object> playerGoals = *new* HashMap<>();  
 ArrayList<Team> teams = *new* ArrayList<>(2);  
 teams.add(*this*);  
 teams.add(visitingTeam);  
 GameCall game = *new* GameCall(d,teams,playerGoals);  
 *this*.games.add(game);  
 *return* game;  
 }  
 *public void* addGame(GameCall gm){  
 *if*(findGame(gm.getDate())!=*null*){  
 *this*.games.add(gm);  
 }  
 }  
 *public* GameCall findGame(Date d){  
 *for* (GameCall g: *this*.games) {  
 *if*(g.getDate().compareTo(d) == 1){  
 *return* g;  
 }  
 }  
 *return null*;  
 }  
 *public* ArrayList<Player> findPlayers(String position){  
 ArrayList<Player> returnP = *new* ArrayList<>(20);  
 *for* (Player p:*this*.players) {  
 *if*(Objects.*equals*(p.getPosition(), position)){  
 returnP.add(p);  
 }  
 }  
 *return* returnP;  
 }  
 *public* Player findPlayer(*int* number){  
 *for* (Player p:*this*.players) {  
 *if*(Objects.*equals*(p.getNumber(),number)){  
 *return* p;  
 }  
 }  
 *return null*;  
 }  
 *public void* hirePlayer(Player p){  
 *this*.players.add(p);  
 }  
 *public* Player sellPlayer(*int* number){  
 Player p = findPlayer(number);  
 *this*.players.remove(number);  
 *return* p;  
 }  
 *public* Player transferPlayer(*int* number,*TeamManagmentI* team){  
 Player p = *null*;  
 *for* (*int* i = 0; i <*this*.players.size() ; i++) {  
 p = *this*.players.get(i);  
 *if*(p.getNumber() == number){  
 team.hirePlayer(p);  
 *this*.sellPlayer(number);  
 *return* p;  
 }  
 }  
 *return null*;  
 }  
  
 *public static void* main(String[] args) {  
 ArrayList<Player> jogadoresP = *new* ArrayList<>();  
 Player p1 = *new* Player("1","Joao","nao sei",*new* Date((*short*) 1,(*short*) 1,(*short*)1),10,"Striker","9","nao joga","nao joga",*false*,5);  
 Team t1 = *new* Team("Porto",jogadoresP);  
 t1.players.add(p1);  
 ArrayList<Player> jogadoresB = *new* ArrayList<>();  
 Player p2 = *new* Player("1","Andre","nao sei",*new* Date((*short*) 1,(*short*) 1,(*short*)1),20,"Midfielder","nao joga","10","nao joga",*false*,5);  
 Team t2 = *new* Team("Boavista",jogadoresB);  
 t2.players.add(p2);  
 GameCall g1 = t1.createGameCall(*new* Date((*short*) 10,(*short*) 10,(*short*)10),t2);  
 t1.games.add(g1);  
 t2.games.add(g1);  
 *for* (*int* i = 0; i < g1.getTeams().size() ; i++) {  
 StdOut.*println*(g1.getTeams().get(i).getName());  
 }  
 */\*for (int i = 0; i <jogadoresP.size() ; i++) {  
 p = jogadoresP.get(i);  
 g1.joinPlayer(p.getNumber(),"Porto");  
 p = jogadoresB.get(i);  
 g1.joinPlayer(p.getNumber(),"Boavista");  
 }\*/* StdOut.*println*(t1.findGame(*new* Date((*short*) 10,(*short*) 10,(*short*)10)).getDate());  
 StdOut.*println*(t1.findPlayers("Striker"));  
 StdOut.*println*(g1.addPlayerGoal(10,"Porto"));  
 t1.hirePlayer(p2);  
 *for* (*int* i = 0; i <jogadoresP.size() ; i++) {  
 StdOut.*println*(t1.players.get(i).toString());  
 }  
  
 }

**Class – Player**

*package* edu.ufp.inf.lp2.Teste;  
  
  
*import* edu.ufp.inf.lp2.intro\_01.Date;  
*import* edu.ufp.inf.lp2.intro\_01.Person;  
  
*public class* Player *extends* Person {  
 *private int* number;  
 *private* String position;  
 *private final* String FORWARDER;  
 *private final* String MIDFIELDER;  
 *private final* String DEFENSER;  
 *private boolean* injured;  
 *private int* goals;  
  
 @Override  
 *public* String toString() {  
 *return* "Player{" +  
 "number=" + number +  
 ", position='" + position + '\'' +  
 ", FORWARDER='" + FORWARDER + '\'' +  
 ", MIDFIELDER='" + MIDFIELDER + '\'' +  
 ", DEFENSER='" + DEFENSER + '\'' +  
 ", injured=" + injured +  
 ", goals=" + goals +  
 '}';  
 }  
  
 *public* Player(String idNumber, String name, String adress, Date birth, *int* number, String position, String FORWARDER, String MIDFIELDER, String DEFENSER, *boolean* injured, *int* goals){  
 *super*(idNumber,name,adress,birth);  
 *this*.number=number;  
 *this*.position=position;  
 *this*.FORWARDER=FORWARDER;  
 *this*.MIDFIELDER=MIDFIELDER;  
 *this*.DEFENSER=DEFENSER;  
 *this*.injured=injured;  
 *this*.goals=goals;  
 }  
 *public int* getNumber(){  
 *return this*.number;  
 }  
 *public void* setNumber(*int* number){  
 *this*.number=number;  
 }  
 *public* String getPosition(){  
 *return this*.position;  
 }  
 *public void* setPosition(String position){  
 *this*.position=position;  
 }  
 *public* String getFORWARDER(){  
 *return this*.FORWARDER;  
 }  
 *public* String getMIDFIELDER(){  
 *return this*.MIDFIELDER;  
 }  
 *public* String getDEFENSER(){  
 *return this*.DEFENSER;  
 }  
 *public boolean* isInjured(){  
 *return this*.injured;  
 }  
 *public void* setInjured(*boolean* injured){  
 *this*.injured=injured;  
 }  
 *public int* getGoals(){  
 *return this*.goals;  
 }  
 *public void* setGoals(*int* goals){  
 *this*.goals=goals;  
 }  
}

**Class – GameCall**

*package* edu.ufp.inf.lp2.Teste;  
  
*import* edu.ufp.inf.lp2.intro\_01.Date;  
  
*import* java.util.ArrayList;  
*import* java.util.HashMap;  
*import* java.util.Objects;  
  
*public class* GameCall {  
 *private* Date date;  
 *private* ArrayList<Team> teams = *new* ArrayList<Team>(2);  
 *private* HashMap<Object, Object> playerGoals = *new* HashMap<>();  
 *private* ArrayList<Player> callHomeTeam = *new* ArrayList<>();  
 *private* ArrayList<Player> callVisitingTeam = *new* ArrayList<>();  
  
 *public* ArrayList<Player> getCallHomeTeam() {  
 *return* callHomeTeam;  
 }  
  
 *public void* setCallHomeTeam(ArrayList<Player> callHomeTeam) {  
 *this*.callHomeTeam = callHomeTeam;  
 }  
  
 *public* ArrayList<Player> getCallVisitingTeam() {  
 *return* callVisitingTeam;  
 }  
  
 *public void* setCallVisitingTeam(ArrayList<Player> callVisitingTeam) {  
 *this*.callVisitingTeam = callVisitingTeam;  
 }  
  
 *public* GameCall(Date date, ArrayList<Team> teams, HashMap<Object, Object> playerGoals) {  
 *this*.date = date;  
 *this*.teams = teams;  
 *this*.playerGoals = playerGoals;  
 }  
  
 *public* Date getDate() {  
 *return* date;  
 }  
  
 *public void* setDate(Date date) {  
 *this*.date = date;  
 }  
  
 *public* ArrayList<Team> getTeams() {  
 *return* teams;  
 }  
  
 *public void* setTeams(ArrayList<Team> teams) {  
 *this*.teams = teams;  
 }  
  
 *public* HashMap<Object, Object> getPlayerGoals() {  
 *return* playerGoals;  
 }  
  
 *public void* setPlayerGoals(HashMap<Object, Object> playerGoals) {  
 *this*.playerGoals = playerGoals;  
 }  
  
 *public* Player joinPlayer(*int* number, String team){  
 *if*(Objects.*equals*(*this*.teams.get(0).getName(), team)){  
 *for* (Player o: *this*.teams.get(0).getPlayers()) {  
 *if*(Objects.*equals*(o.getNumber(),number)){  
 *this*.callHomeTeam.add(o);  
 *return* o;  
 }  
 }  
 } *else if* (Objects.*equals*(*this*.teams.get(1).getName(), team)) {  
 *for* (Player o: *this*.teams.get(1).getPlayers()) {  
 *if*(Objects.*equals*(o.getNumber(),number)){  
 *this*.callVisitingTeam.add(o);  
 *return* o;  
 }  
 }  
 }  
 *return null*;  
 }  
 *public boolean* addPlayerGoal(*int* number,String team){  
 Player player = joinPlayer(number,team);  
 *if*(player != *null*){  
 player.setGoals(player.getGoals() + 1 );  
 *//this.playerGoals.put(player,1);  
 return true*;  
 }  
 *return false*;  
 }  
  
 *// NAO MEXEMOS COM HASMAPS AINDA POR ISSO NAO FIZ  
 /\* public ArrayList<Player> findPlayerScoredMoreGoals(){  
 playerGoals.forEach((key,value) -> {  
  
 });*

**Código GestaoLivros**

**Class – BD**

package GestaoLivros;

import java.util.ArrayList;

public class BD implements GestaoUtilizadores, GestaoLivros{

private ArrayList<Utilizador> utilizadores;

private ArrayList<Livro> livros;

@Override

public void add(Livro l) {

this.livros.add(l);

}

@Override

public Livro procuraLivro(String t) {

for (Livro l: this.livros) {

if(l.getTitulo().equals(t)) return l;

}

return null;

}

@Override

public ArrayList<Livro> procuraLivros(String t, String a) {

ArrayList<Livro> livros = new ArrayList<Livro>();

for (Livro l: this.livros) {

if(l.getTitulo().equals(t)){

for (String autor: l.getAutores()) {

if(autor.equals(a)) livros.add(l);

}

}

}

if(livros.isEmpty()){

return null;

}

return livros;

}

@Override

public void regista(Utilizador u) {

this.utilizadores.add(u);

}

@Override

public boolean existe(Utilizador u) {

if(this.utilizadores == null) return false;

for (Utilizador util: this.utilizadores) {

if(util.equals(u)) return true;

}

return false;

}

public BD(ArrayList<Utilizador> utilizadores, ArrayList<Livro> livros) {

this.utilizadores = utilizadores;

this.livros = livros;

}

public ArrayList<Utilizador> getUtilizadores() {

return utilizadores;

}

public void setUtilizadores(ArrayList<Utilizador> utilizadores) {

this.utilizadores = utilizadores;

}

public ArrayList<Livro> getLivros() {

return livros;

}

public void setLivros(ArrayList<Livro> livros) {

this.livros = livros;

}

}

**Class – GestaoAcesso**

package GestaoLivros;

import Person.Date;

import edu.princeton.cs.algs4.StdOut;

import java.util.ArrayList;

public class GestaoAcesso {

private BD gu;

public boolean registo(Utilizador u){

this.gu.regista(u);

return true;

}

public Sessao login(Utilizador u){

if(this.gu.existe(u)){

return new Sessao(gu);

}

return null;

}

public GestaoAcesso(BD gu) {

this.gu = gu;

}

public static void main(String[] args) {

Date d = new Date((short) 1, (short) 1, (short)1);

BD bd = new BD(new ArrayList<Utilizador>(), new ArrayList<Livro>());

Utilizador u1 = new Utilizador("t", "t", new ArrayList<Livro>());

Utilizador u2 = new Utilizador("a", "a", new ArrayList<Livro>());

bd.regista(u1);

bd.regista(u2);

GestaoAcesso ga = new GestaoAcesso(bd);

Sessao s1 = ga.login(u1);

ArrayList<String> autores = new ArrayList<>(2);

autores.add("Jose");

autores.add("dhaiujdhawoi");

Livro l1 = new Livro("a1", autores, d);

Livro l2 = new Livro("b1", autores, d);

Livro l3 = new Livro("c1", autores, d);

ArrayList<Livro> arrL = new ArrayList<Livro>();

arrL.add(l1);

arrL.add(l2);

arrL.add(l3);

for (int i = 0; i < arrL.size(); i++){

s1.adicionaLivro(l1);

}

s1.procuraLivros(l1.getTitulo(), l1.getAutores().get(0));

s1.requisitaLivro(l1, u1.getUname());

StdOut.println(s1.procuraLivros(l1.getTitulo(), l1.getAutores().get(0)));

s1.esprestaLivro(l1, u1.getUname(), u2.getUname());

}

}

**Class – GestaoLivro**

package GestaoLivros;

import java.util.ArrayList;

public interface GestaoLivros {

public void add(Livro l);

public Livro procuraLivro(String t);

public ArrayList<Livro> procuraLivros(String t, String a);

}

**Class – GestaoUtilizadores**

package GestaoLivros;

public interface GestaoUtilizadores {

public void regista(Utilizador u);

public boolean existe(Utilizador u);

}

**Class – Livro**

package GestaoLivros;

import Person.\*;

import java.util.ArrayList;

public class Livro {

private String titulo;

private ArrayList<String> autores;

private Date publicacao;

private Utilizador utilizador;

public boolean requisitado(){

return this.getUtilizador() != null;

}

public Livro(String titulo, ArrayList<String> autores, Date publicacao) {

this.titulo = titulo;

this.autores = autores;

this.publicacao = publicacao;

}

public String getTitulo() {

return titulo;

}

public void setTitulo(String titulo) {

this.titulo = titulo;

}

public ArrayList<String> getAutores() {

return autores;

}

public void setAutores(ArrayList<String> autores) {

this.autores = autores;

}

public Date getPublicacao() {

return publicacao;

}

public void setPublicacao(Date publicacao) {

this.publicacao = publicacao;

}

public Utilizador getUtilizador() {

return utilizador;

}

public void setUtilizador(Utilizador utilizador) {

this.utilizador = utilizador;

}

}

**Class – Sessão**

package GestaoLivros;

import java.util.ArrayList;

public class Sessao {

private BD gl;

public void adicionaLivro(Livro b){

this.gl.add(b);

}

public ArrayList<Livro> procuraLivros(String t, String a){

return this.gl.procuraLivros(t,a);

}

public void requisitaLivro(Livro l, String u){

if(!l.requisitado()){

for (Utilizador util: this.gl.getUtilizadores()) {

if(util.getUname().equals(u)){

util.recebe(l);

l.setUtilizador(util);

}

}

}

}

public void esprestaLivro(Livro l, String u1, String u2){

if(l.requisitado()){

for (Utilizador util1: this.gl.getUtilizadores()) {

if(util1.getUname().equals(u1)){

for (Utilizador util2: this.gl.getUtilizadores()) {

if(util2.getUname().equals(u2)){

util1.empresta(l, util2);

}

}

}

}

}

}

public Sessao(BD gl) {

this.gl = gl;

}

}

**Class – Utilizador**

package GestaoLivros;

import Person.\*;

import java.util.ArrayList;

public class Utilizador {

private String uname;

private String passwd;

private ArrayList<Livro> livros;

public void recebe(Livro l){

this.livros.add(l);

}

public Livro devolve(String t){

for (Livro l: this.livros) {

if(l.getTitulo().equals(t)) {

this.livros.remove(l);

return l;

}

}

return null;

}

public void empresta(Livro l, Utilizador u){

u.livros.add(l);

this.livros.remove(l);

}

public Utilizador(String uname, String passwd, ArrayList<Livro> livros) {

this.uname = uname;

this.passwd = passwd;

this.livros = livros;

}

public String getUname() {

return uname;

}

public void setUname(String uname) {

this.uname = uname;

}

public String getPasswd() {

return passwd;

}

public void setPasswd(String passwd) {

this.passwd = passwd;

}

public ArrayList<Livro> getLivros() {

return livros;

}

public void setLivros(ArrayList<Livro> livros) {

this.livros = livros;

}

}

**Class – Utilizador Inexistente**

package GestaoLivros;

public class UtilizadorInexistenteExecption extends Exception{

public UtilizadorInexistenteExecption() {

}

public UtilizadorInexistenteExecption(String message) {

super(message);

}

public UtilizadorInexistenteExecption(String message, Throwable cause) {

super(message, cause);

}

public UtilizadorInexistenteExecption(Throwable cause) {

super(cause);

}

public UtilizadorInexistenteExecption(String message, Throwable cause, boolean enableSuppression, boolean writableStackTrace) {

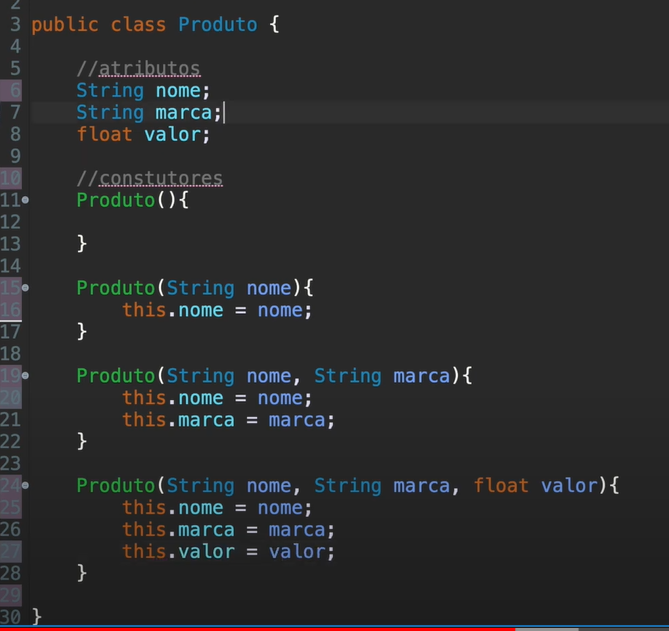
super(message, cause, enableSuppression, writableStackTrace);

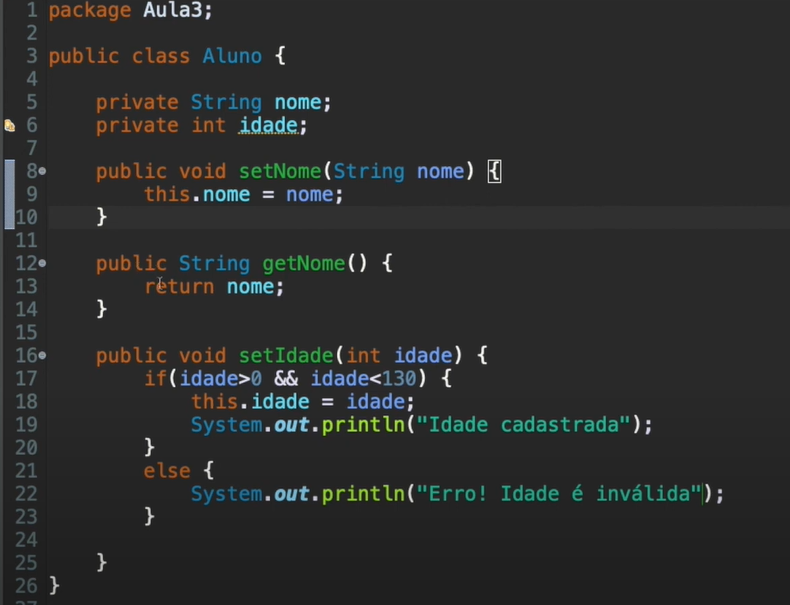
}

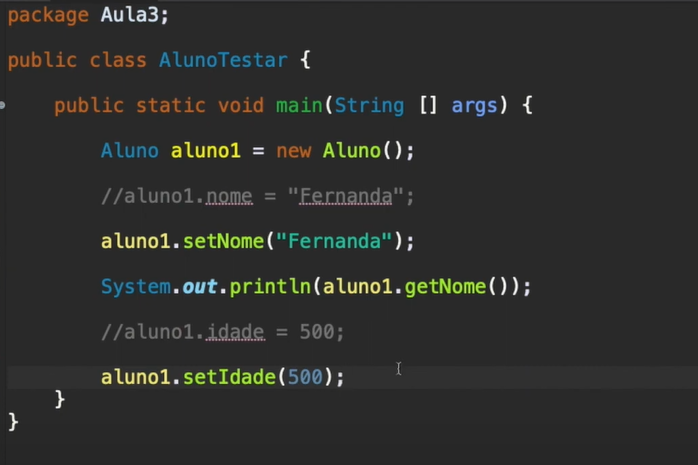
}

**Exemplos da Nanda**

**Construtores**

****

**Getters & Setters**



**ArrayLists**

Uma imagem com texto

Descrição gerada automaticamente