publico classe SistemaControleTreino {

publico estatico método principal(Caractere[] args) {

Aluno a[] = novo Aluno[2];

a[0] = novo Aluno(1234,"Iniciante");

a[1] = novo Aluno(1234,"Iniciante");

Instrutor i[] = novo Instrutor[2];

i[0] = novo Instrutor("Musculação",4500.00f, verdadeiro);

i[1] = novo Instrutor("Musculação",4500.00f, falso);

Recepcionista r[] = novo Recepcionista[2];

r[0] = novo Recepcionista(falso);

r[1] = novo Recepcionista(verdadeiro);

Treino UTI01 = novo Treino();

UTI01.marcarTreino(i[0],a[1] );

UTI01.Treinar();

}

}

classe abstrata Pessoa

privado nome: caractere

privado idade: inteiro

privado nome: caractere

publico metodo fazeraniversario()

idade++

fimMetodo

fimclasse

+getters

+setters

---------------------------------------------------------------------------------------------------------------------

classe Aluno estende Pessoa

privado matricula: inteiro

privado treino: caractere

publico Aluno(inteiro matricula, caractere treino) {

this.matricula = matricula

this.treino = treino

}fimConstrutor

publico metodo cancelarMatricula()

x:caractere

Scanner sc = novo Scanner(System.in)

escreva("Tem certeza que deseja cancelar sua matrícula?")

leia(x)

se(x.igual("sim")){

escreva("matrícula cancelada com sucesso" + getNome")

}senão{

escreva("Não entendi. Por favor, fale com um de nossos recepcionistas."

}fimSe

fimMetodo

+getters

+setters

---------------------------------------------------------------------------------------------------------------------

classe Instrutor estende Pessoa

privado especialidade: String

privado salario: real

privado presente: boleano

publico Instrutor(caractere especialidade, real salario, boleano presente) {

this.especialidade = especialidade

this.salario = salario

this.presente = presente

}fimconstrutor

publico metodo darAula(boleano recepcionista)

se(recepcionista){

escreva("Preciso checar minha agenda de treinos...")

}fimSE

fimMetodo

publico metodo darAula(caractere fala)

se(fala.igual("estou pronto") || fala.igual("olá")){

escreva("Pegar ficha de Treino")

}senao{

escreva("Falar com outro aluno")

}fimSe

fimMetodo

publico metodo darAula(inteiro hora, inteiro minuto)

se(hora > 8 && hora < 12 ){

escreva("Bom dia! Pronto pra se fMalhar?")

}senao se( hora >=12 && hora <14){

this.presente = !this.presente

escreva("Estou indo almoçar... Depois nos falamos!")

}senao se(hora >=14 && hora < 17){

escreva("Boa tarde! Bora treinar...")

}senao{

escreva("indisponível")

this.presente = !this.presente

}fimSe

fimMetodo

public metodo receberAumento(real aum)

this.salario += aum

fimMetodo

+getters

+setters

---------------------------------------------------------------------------------------------------------------------

publico class Recepcionista estende Pessoa {

publico boleano trabalhando

publico Recepcionista(boleano trabalhando) {

this.trabalhando = trabalhando

}fimConstrutor

publico metodo mudaTrabalho()

this.trabalhando = !this.trabalhando

fimMetodo

+getters

+setters

---------------------------------------------------------------------------------------------------------------------

publico class Treino {

privado Aluno cliente;

privado Instrutor treinador;

privado booleano marcado;

privado inteiro pergunta;

publico caractere treinoH[];

publico caractere treinoE[];

publico inteiro aluno;

publico metodo marcarTreino(Instrutor i, Aluno a)

{

se (i.getPresente() == verdadeiro) {

se (this.marcado == falso) {

Scanner sc = novo Scanner(System.in);

EscrevaL("Qual seu objetivo? \n "

+ "------------------------\n"

+ "1-Emagrecimento \n"

+ "2-Ganho de Massa Muscular \n");

leia (pergunta = sc.nextInt();)

se(pergunta == 1) {

EscrevaL("Treino para Emagrecimento criado!");

this.marcado = verdadeiro;

this.cliente = a;

this.treinador = i;

EscrevaL("Treino marcado!");

treinoE = novo Caractere[3];

treinoE[0] = "A";

treinoE[1] = "B";

treinoE[2] = "C";

} senao se (pergunta == 2) {

EscrevaL("Treino para Hipertrofia criado!");

this.marcado = verdadeiro;

this.cliente = a;

this.treinador = i;

EscrevaL("Treino marcado!");

treinoH = novo Caractere[3];

treinoH[0] = "A";

treinoH[1] = "B";

treinoH[2] = "C";

} senao{

EscrevaL("opção invalida!");

}

FimSe

}

} senao{

this.marcado = falso;

this.cliente = null;

this.treinador = null;

System.out.println("Impossibilitado de marcar treino");

}

FimSe

}

fimMetodo

publico metodo Treinar() {

se (this.marcado) {

para (intiro treinos = 0; treinos <= 10; treinos++) {

se (treinos < 10) {

se (pergunta == 1) {

para (inteiro i = 0; i < treinoE.length; i++) {

escrevaL("------------------------------")

escrevaL("Hoje é o seu treino: " + treinoE[i])

escrevaL("Hoje é o treino de numero: " + treinos + " faltam "

+ (10 - treinos) + " treinos para trocar o seu treino")

}

} senao se(pergunta == 2) {

para (inteiro i = 0; i < treinoH.length; i++) {

escreva("------------------------------")

escreva("Hoje é o seu treino: " + treinoH[i])

escreva("Hoje é o treino de numero: " + treinos

+ " de treino faltam \n" + (10 - treinos)

+ " treinos para trocar o seu treino")

}

}

} senao {

escreva("Seu treino expirou! \n Favor agendar um novo treino!!!")

}FimSe

}

} senao {

escreva("O treino não pode acontecer...")

}FimSe

}fimMetodo

+getters

+setters