Códigos:

package com.mycompany.petshop.model;

import java.util.ArrayList;

import java.util.Date;

/\*\*

\*

\* @author dti

\*/

// Classe Persistencia (projeto Sistema de Agendamentos para PetShop)

public class Persistencia {

private static ArrayList<Usuario> listaUsuarios = new ArrayList<Usuario>();

private static ArrayList<Servico> listaServicos = new ArrayList<Servico>();

private static ArrayList<Produto> listaProdutos = new ArrayList<Produto>();

private static ArrayList<Pet> listaPets = new ArrayList<Pet>();

private static ArrayList<Agendamento> listaAgendamentos = new ArrayList<Agendamento>();

// Usuários

static {

// Usuário admin criado ao iniciar o sistema

Usuario admin = new Usuario("Administrador", "admin", "00000000000", "admin", "ADMIN");

listaUsuarios.add(admin);

}

// Método que retorna a lista de usuários

public static ArrayList<Usuario> getLista() {

return listaUsuarios;

}

// Método que salva o usuário cadastrado na lista de usuários

public static void cadastrar(Usuario user) {

if (user != null) {

listaUsuarios.add(user);

}

}

// Método que busca o usuário pelo seu email cadastrado

public static Usuario buscarPorEmail(String email) {

for (Usuario u : listaUsuarios) {

if (u.getEmail().equalsIgnoreCase(email)) {

return u;

}

}

return null;

}

// Método que busca o usuário pelo seu CPF cadastrado

public static Usuario buscarPorCpf(String cpf) {

for (Usuario u : listaUsuarios) {

if (u.getCpf().equals(cpf)) {

return u;

}

}

return null;

}

// Método que verifica que o usuário está na lista de usuários para conseguir autentica-lo, retornando o usuário

public static Usuario autenticar(String email, String senha) {

for (Usuario u : listaUsuarios) {

if (u.getEmail().equalsIgnoreCase(email) && u.getSenha().equals(senha)) {

return u;

}

}

return null;

}

// Método que busca o usuário pelo seu id

public static Usuario getUsuarioPorId(int id) {

for (Usuario u : listaUsuarios) {

if (u.getId() == id) {

return u;

}

}

return null; // Não encontrado

}

// Serviços

// Método que salva o serviço na lista de serviços

public static void cadastrarServico(Servico s) {

if (s != null) {

listaServicos.add(s);

}

}

// Método que retorna a lista de serviços

public static ArrayList<Servico> getServicos() {

return listaServicos;

}

// Produtos

// Método que salva o produto na lista de produtos

public static void cadastrarProduto(Produto p) {

if (p != null) {

listaProdutos.add(p);

}

}

// Método que retorna a lista de Produtos

public static ArrayList<Produto> getProdutos() {

return listaProdutos;

}

// Pets

// Método que salva o pet na lista de Pets

public static void cadastrarPet(Pet pet) {

if (pet != null) {

listaPets.add(pet);

}

}

// Método que retorna a lista de Pets

public static ArrayList<Pet> getPets() {

return listaPets;

}

// Método que busca o pet pelo seu nome

public static Pet buscarPetPorNome(String nome) {

for (Pet p : listaPets) {

if (p.getNome().equalsIgnoreCase(nome)) {

return p;

}

}

return null;

}

// Método que busca o pet pelo seu id

public static Pet getPetPorId(int id) {

for (Pet p : listaPets) {

if (p.getId() == id) {

return p;

}

}

return null; // Não encontrado

}

// Agendamentos

// Método que retorna a lista de Agendamentos

public static ArrayList<Agendamento> getAgendamentos() {

return listaAgendamentos;

}

// Método que verifica se há algum conflito na data e hora no momento em que é feita a tentativa de salvar uma agendamento

public static boolean temConflito(Date inicio, int duracaoMinutos) {

long inicioMillis = inicio.getTime();

long fimMillis = inicioMillis + (duracaoMinutos \* 60 \* 1000);

for (Agendamento a : listaAgendamentos) {

long agendamentoInicio = a.getDataHora().getTime();

long agendamentoFim = a.getFimAgendamento().getTime();

// Conflito: início < fim de agendamento existente E fim > início de agendamento existente

if (inicioMillis < agendamentoFim && fimMillis > agendamentoInicio) {

return true;

}

}

return false;

}

// Método que salva o agendamento na lista de agendmanetos se a data e hora não tiver nenhum conflito

public static boolean cadastrarAgendamento(Agendamento agendamento) {

if (!temConflito(agendamento.getDataHora(), agendamento.getTempoTotal())) {

listaAgendamentos.add(agendamento);

return true;

}

return false; // Conflito encontrado, não cadastra

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

package com.mycompany.petshop.model;

// Classe Usuario (projeto Sistema de Agendamentos para PetShop)

public class Usuario {

private static int proximoId = 1;

private int id;

private String nome;

private String email;

private String cpf;

private String senha;

private String role;

// Construtor

public Usuario(String nome, String email, String cpf, String senha, String role) {

this.id = proximoId++;

this.nome = nome;

this.email = email;

this.cpf = cpf;

this.senha = senha;

this.role = role;

}

// Getters e Setters

public static int getProximoId() {

return proximoId;

}

public static void setProximoId(int proximoId) {

Usuario.proximoId = proximoId;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getNome() {

return nome;

}

public void setNome(String nome) {

this.nome = nome;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getCpf() {

return cpf;

}

public void setCpf(String cpf) {

this.cpf = cpf;

}

public String getSenha() {

return senha;

}

public void setSenha(String senha) {

this.senha = senha;

}

public String getRole() {

return role;

}

public void setRole(String role) {

this.role = role;

}

@Override

public String toString() {

return "Usuario{" + "id=" + id + ", nome=" + nome + ", email=" + email + ", cpf=" + cpf + ", senha=" + senha + '}';

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

package com.mycompany.petshop.model;

// Classe Servico (projeto Sistema de Agendamentos para PetShop)

public class Servico {

private static int proximoId = 1;

private int id;

private String nome;

private float preco;

private int tempoEstimado; // em minutos

// Construtor

public Servico(String nome, float preco, int tempoEstimado) {

this.id = proximoId++;

this.nome = nome;

this.preco = preco;

this.tempoEstimado = tempoEstimado;

}

// Getters e Setters

public static int getProximoId() {

return proximoId;

}

public static void setProximoId(int proximoId) {

Servico.proximoId = proximoId;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getNome() {

return nome;

}

public void setNome(String nome) {

this.nome = nome;

}

public float getPreco() {

return preco;

}

public void setPreco(float preco) {

this.preco = preco;

}

public float getTempoEstimado() {

return tempoEstimado;

}

public void setTempoEstimado(int tempoEstimado) {

this.tempoEstimado = tempoEstimado;

}

@Override

public String toString() {

return "ID: " + id + " | Nome: " + nome + " | Preço: R$" + preco + " | Tempo: " + tempoEstimado + " min";

}

}

package com.mycompany.petshop.model;

// Classe Produto (projeto Sistema de Agendamentos para PetShop)

public class Produto {

private static int proximoId = 1;

private int id;

private String nome;

private float preco;

private int quantidade;

private String categoria;

private String descricao;

// Construtor

public Produto(String nome, float preco, int quantidade, String categoria, String descricao) {

this.id = proximoId++;

this.nome = nome;

this.preco = preco;

this.quantidade = quantidade;

this.categoria = categoria;

this.descricao = descricao;

}

// Getters e Setters

public static int getProximoId() {

return proximoId;

}

public static void setProximoId(int proximoId) {

Produto.proximoId = proximoId;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getNome() {

return nome;

}

public void setNome(String nome) {

this.nome = nome;

}

public float getPreco() {

return preco;

}

public void setPreco(float preco) {

this.preco = preco;

}

public float getQuantidade() {

return quantidade;

}

public void setQuantidade(int quantidade) {

this.quantidade = quantidade;

}

public String getCategoria() {

return categoria;

}

public void setCategoria(String categoria) {

this.categoria = categoria;

}

public String getDescricao() {

return descricao;

}

public void setDescricao(String descricao) {

this.descricao = descricao;

}

@Override

public String toString() {

return "Produto{" + "id=" + id + ", nome=" + nome + ", preco=" + preco + ", quantidade=" + quantidade + ", categoria=" + categoria + ", descricao=" + descricao + '}';

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

package com.mycompany.petshop.model;

// Classe Pet (projeto Sistema de Agendamentos para PetShop)

public class Pet {

private static int proximoId = 1;

private int id;

private int usuarioId;

private String nome;

private String porte;

private String idade;

private String raca;

private String observacoes;

// Construtor

public Pet(int usuarioId, String nome, String porte, String idade, String raca, String observacoes) {

this.id = proximoId++;

this.usuarioId = usuarioId;

this.nome = nome;

this.porte = porte;

this.idade = idade;

this.raca = raca;

this.observacoes = observacoes;

}

// Getters e Setters

public static int getProximoId() {

return proximoId;

}

public static void setProximoId(int proximoId) {

Pet.proximoId = proximoId;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getUsuarioId() {

return usuarioId;

}

public void setUsuarioId(int usuarioId) {

this.usuarioId = usuarioId;

}

public String getNome() {

return nome;

}

public void setNome(String nome) {

this.nome = nome;

}

public String getPorte() {

return porte;

}

public void setPorte(String porte) {

this.porte = porte;

}

public String getIdade() {

return idade;

}

public void setIdade(String idade) {

this.idade = idade;

}

public String getRaca() {

return raca;

}

public void setRaca(String raca) {

this.raca = raca;

}

public String getObservacoes() {

return observacoes;

}

public void setObservacoes(String observacoes) {

this.observacoes = observacoes;

}

@Override

public String toString() {

return "Pet{" + "id=" + id + ", usuarioId=" + usuarioId + ", nome=" + nome + ", porte=" + porte + ", idade=" + idade + ", raca=" + raca + ", observacoes=" + observacoes + '}';

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

package com.mycompany.petshop.model;

import java.util.ArrayList;

import java.util.Date;

// Classe Agendamento (projeto Sistema de Agendamentos para PetShop)

public class Agendamento {

private static int proximoId = 1;

private int id;

private int usuarioId;

private int petId;

private ArrayList<Servico> servicos;

private Date dataHora; // Data e hora do agendamento

private int tempoTotal; // Em minutos

private double valorTotal;

// Construtor

public Agendamento(int usuarioId, int petId, ArrayList<Servico> servicos, Date dataHora) {

this.id = proximoId++;

this.usuarioId = usuarioId;

this.petId = petId;

this.servicos = servicos;

this.dataHora = dataHora;

this.tempoTotal = calcularTempoTotal();

this.valorTotal = calcularValorTotal();

}

// Função para calcular o tempo estimado total dos serviços no agendamento

private int calcularTempoTotal() {

int total = 0;

for (Servico s : servicos) {

total += s.getTempoEstimado();

}

return total;

}

// Função para cacular o valor total dos serviços no agendamento

private double calcularValorTotal() {

double total = 0.0;

for (Servico s : servicos) {

total += s.getPreco();

}

return total;

}

// Getters e Setters

public static int getProximoId() {

return proximoId;

}

public static void setProximoId(int proximoId) {

Agendamento.proximoId = proximoId;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getUsuarioId() {

return usuarioId;

}

public void setUsuarioId(int usuarioId) {

this.usuarioId = usuarioId;

}

public int getPetId() {

return petId;

}

public void setPetId(int petId) {

this.petId = petId;

}

public ArrayList<Servico> getServicos() {

return servicos;

}

public void setServicos(ArrayList<Servico> servicos) {

this.servicos = servicos;

}

public Date getDataHora() {

return dataHora;

}

public void setDataHora(Date dataHora) {

this.dataHora = dataHora;

}

public int getTempoTotal() {

return tempoTotal;

}

public void setTempoTotal(int tempoTotal) {

this.tempoTotal = tempoTotal;

}

public double getValorTotal() {

return valorTotal;

}

public void setValorTotal(double valorTotal) {

this.valorTotal = valorTotal;

}

public Date getFimAgendamento() {

long millis = dataHora.getTime() + (tempoTotal \* 60 \* 1000);

return new Date(millis);

}

public Object getUsuario() {

throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}

}

Princípios de POO utilizados:

Encapsulamento: Uso de private em todas as classes devido a cada cliente ter seus dados individuais e todas as classes possuírem get/set para possíveis visualizações e alterações nos dados.

Herança: Não aplicável

Polimorfismo: Temos os métodos toString para todas as classes, mas cada um corresponde a própria classe de sua maneira, embora o mesmo nome.

Classes e Objetos:

. Agendamento: Classe que representa um horário marcado, assim como se refere a pessoa a qual fez o agendamento e sobre qual pet é o agendamento no sistema. A classe também proporciona o método de cálculo de tempo e valor total, demonstrando como o sistema proporciona visualizações dos dados para maior entendimento do administrador.

. Persistência: Classe que representa a persistência dos dados do sistema, responsável pelas suas funcionalidades relacionados a dados. Serviços como autenticar e buscar por e-mail e cpf estão nessa classe, demonstrando encapsulamento sobre métodos que outras classes não podem ter acesso.

. Pet: Classe que representa os animais de cada cliente no sistema e seus atributos básicos, assim como gets e sets.

. Produto: Classe que representa os produtos do sistema e seus atributos básicos, assim como gets e sets.

. Serviço: Classe que representa os tipos de serviços disponíveis no sistema, diretamente ligado aos serviços disponíveis para agendamentos, assim como gets e sets.

. Usuário: Classe que representa o cliente/usuário do sistema com seus atributos básicos, assim como gets e sets.