Nome: Igor Moreira Pádua
Matrícula: 202009567

Exercício 1:

```
public class Exercicio1 {
	public static void main(String[] args) {
		Funcionarios[] funcionario = new Funcionarios[10];

		// Gerente:
		funcionario[0] = new Gerente("Roberto", "Silva");

		((Gerente) funcionario[0]).setSalario(2500);
		((Gerente) funcionario[0]).setBonificacao(150);

		// Horistas:
		funcionario[1] = new Horistas("Luiza", "Rodrigues");

		((Horistas) funcionario[1]).setHorasTrabalhadas(80);
		((Horistas) funcionario[1]).setValorHora(15);

		funcionario[2] = new Horistas("Patrick", "Alvez");

		((Horistas) funcionario[2]).setHorasTrabalhadas(75);
		((Horistas) funcionario[2]).setValorHora(12);

		// Comissionados
		funcionario[3] = new Comissionados("Felipe", "Mendez");

		((Comissionados) funcionario[3]).setSalario(1200);
		((Comissionados) funcionario[3]).setTotalVendas(100);
		((Comissionados) funcionario[3]).setPorcentualComissao(12);

		funcionario[4] = new Comissionados("Santiago", "Cardoso");

		((Comissionados) funcionario[4]).setSalario(1250);
		((Comissionados) funcionario[4]).setTotalVendas(93);
		((Comissionados) funcionario[4]).setPorcentualComissao(10);

		funcionario[5] = new Comissionados("Marcelo", "Junior");

		((Comissionados) funcionario[5]).setSalario(1030.12);
		((Comissionados) funcionario[5]).setTotalVendas(120);
		((Comissionados) funcionario[5]).setPorcentualComissao(15);

		// Administradores
		funcionario[6] = new Administradores("Diana", "Melo");

		((Administradores) funcionario[6]).setSalario(1950.92);

		funcionario[7] = new Administradores("Rodrigo", "Maia");

		((Administradores) funcionario[7]).setSalario(2000.50);
```

```
                funcionario[8] = new Administradores("Ana", "Julia");

                ((Administradores) funcionario[8]).setSalario(1823);

                funcionario[9] = new Administradores("Fabiano", "Abreu");

                ((Administradores) funcionario[9]).setSalario(2150.56);

                for (int i = 0; i < 10; i++) {
                        System.out.println(funcionario[i]);
                        funcionario[i].mostraSalario();
                }

        }
}
abstract class Funcionarios {

        protected String nome;
        protected String sobrenome;

        Funcionarios(String nome, String sobrenome) {
                this.nome = nome;
                this.sobrenome = sobrenome;
        }

        public abstract void mostraSalario();

        public String toString() {
                return "Nome: " + nome +
                        " - Sobrenome: " + sobrenome;
        }
}

class Administradores extends Funcionarios {

        private double salario;

        Administradores(String nome, String sobrenome) {
                super(nome, sobrenome);
        }

        public void mostraSalario() {
                System.out.format("O salário mensal é R$ %.2f\n", salario);
        }

        public double getSalario() {
                return salario;
        }

        public void setSalario(double salario) {
                this.salario = salario;
        }
```

```
	public String toString() {
		return super.toString() + " - Salario: " + salario;
	}
}

class Gerente extends Funcionarios {

	private double salario;
	private double bonificacao;

	Gerente(String nome, String sobrenome) {
		super(nome, sobrenome);
	}

	public double getBonificacao() {
		return bonificacao;
	}

	public double getSalario() {
		return salario;
	}

	public void setBonificacao(double bonificacao) {
		this.bonificacao = bonificacao;
	}

	public void setSalario(double salario) {
		this.salario = salario;
	}

	public void mostraSalario() {
		System.out.format("O salário mensal é R$ %.2f\n", salario + bonificacao);
	}

	public String toString() {
		return super.toString() + " - Salario: " + salario +
				" - Bonificação: " + bonificacao;
	}
}

class Comissionados extends Funcionarios {

	private double salario;
	private int totalVendas;
	private double porcentualComissao;

	Comissionados(String nome, String sobrenome) {
		super(nome, sobrenome);
	}

	public double getSalario() {
		return salario;
	}
```

```
	public int getTotalVendas() {
		return totalVendas;
	}

	public double getPorcentualComissao() {
		return porcentualComissao;
	}

	public void setSalario(double salario) {
		this.salario = salario;
	}

	public void setTotalVendas(int totalVendas) {
		this.totalVendas = totalVendas;
	}

	public void setPorcentualComissao(double porcentualComissao) {
		this.porcentualComissao = porcentualComissao;
	}

	public void mostraSalario() {
		System.out.format("O sálario é R$ %.2f\n", salario + (totalVendas *
(porcentualComissao / 100)));
	}

	public String toString() {
		return super.toString() + " - Sálario: " + salario +
				" - Total de Vendas: " + totalVendas +
				" - Porcentual de comissão: " + porcentualComissao;
	}
}

class Horistas extends Funcionarios {

	private double horasTrabalhadas;
	private double valorHora;

	Horistas(String nome, String sobrenome) {
		super(nome, sobrenome);
	}

	public double getHorasTrabalhadas() {
		return horasTrabalhadas;
	}

	public double getValorHora() {
		return valorHora;
	}

	public void setHorasTrabalhadas(double horasTrabalhadas) {
		this.horasTrabalhadas = horasTrabalhadas;
	}

	public void setValorHora(double valorHora) {
```

```
		this.valorHora = valorHora;
	}

	public void mostraSalario() {
		System.out.format("O sálario é R$ %.2f\n", horasTrabalhadas * valorHora);
	}

	public String toString() {
		return super.toString() + " - Horas trabalhadas:" + horasTrabalhadas +
				" - Valor da hora:" + valorHora;
	}
}
```

Exercício 2:

```
public class Exercicio2 {

	public static void main(String[] args) {


		// Matriz:

		LojaConcreta matriz = new LojaConcreta("23748234", "Indrustria peças");


		matriz.endereco = "Rua 912, qt 99";

		matriz.gerente = "Lucas";

		matriz.identificador = 162;

		matriz.registra_abertura_dia();


		// Filial 1:

		LojaConcreta filial1 = new LojaConcreta("435212", "Vendas de peças");


		filial1.endereco = "Rua 521, qt 85";

		filial1.gerente = "Roberto";

		filial1.identificador = 563;

		filial1.registra_abertura_dia();


		// Filial 2:

		LojaConcreta filial2 = new LojaConcreta("882340", "Compra de peças
usadas");


		filial2.endereco = "Rua 198, qt 23";
```

```
                filial2.gerente = "Sabrina";
                filial2.identificador = 329;
                filial2.registra_fechamento_dia();

                // Fechamento da matriz
                matriz.registra_fechamento_dia();

                System.out.println(matriz);
                System.out.println(filial1);
                System.out.println(filial2);
        }
}

abstract class Loja {

        protected int identificador;
        protected String cnpj;
        protected String razaoSocial;
        protected boolean aberta;

        Loja(String cnpj, String razaoSocial) {
                this.cnpj = cnpj;
                this.razaoSocial = razaoSocial;
                aberta = false;
        }

        public String getCnpj() {
                return cnpj;
        }

        public String getRazaoSocial() {
                return razaoSocial;
        }
```

```
	public boolean getAberta() {
		return aberta;
	}

	public String toString() {
		return "Identificador: " + identificador +
			" - Cnpj: " + cnpj +
			" - Razão social: " + razaoSocial +
			" - Aberta: " + aberta;
	}
}

interface Registro {
	public void registra_abertura_dia();
	public void registra_fechamento_dia();
}

class LojaConcreta extends Loja implements Registro {

	protected String endereco;
	protected String gerente;

	LojaConcreta(String cnpj, String razaoSocial) {
		super(cnpj, razaoSocial);
	}

	public void registra_abertura_dia() {
		if (this.aberta == true) {
			System.out.format("A loja %d já está aberta.\n",
this.identificador);
		} else {
			this.aberta = true;
```

```
                        System.out.format("A loja %d foi aberta.\n", this.identificador);
                }
        }

        public void registra_fechamento_dia() {
                if (this.aberta == false) {
                        System.out.format("A loja %d já está fechada.\n",
this.identificador);
                } else {
                        this.aberta = false;
                        System.out.format("A loja %d foi fechada.\n",
this.identificador);
                }
        }

        public String toString() {
                return super.toString() +
                        " - Endereço: " + endereco +
                        " - Gerente: " + gerente;
        }
}
```

Exercício 3:

```
import java.util.ArrayList;

public class Exercicio3 {
        public static void main(String[] args) {

                ArrayList<Pessoa> pessoa = new ArrayList<>();

                // Aparelho 1:
                Aparelho ap1 = new Aparelho("4326", 2018, "Samsung", "A320");
                Atendimento aten1 = new Atendimento("12/02/2020", "Dividiu em 3x",
ap1);
```

```
                        // Adicionando um atendimento ao aparelho 1
                        ap1.adicionaAtendimento(aten1);
                        Funcionario fun1 = new Funcionario("Daniel", "235827384", 5321, aten1);
                        // Dono do aparelho ap1
                        Cliente cli1 = new Cliente("Roberto", "23478231", "6282439813", ap1);
                        pessoa.add(cli1);
                        pessoa.add(fun1);

                        // Aparelho 2:
                        Aparelho ap2 = new Aparelho("23952", 2019, "Xiomi", "23X");
                        Atendimento aten2 = new Atendimento("26/08/2019", "Comprou a vista",
ap2);
                        // Adicionando um atendimento ao aparelho 2
                        ap2.adicionaAtendimento(aten2);
                        Funcionario fun2 = new Funcionario("Jorges", "3589243", 1831, aten2);
                        // Dono do aparelho ap2
                        Cliente cli2 = new Cliente("Daniela", "9384521", "629342623", ap2);
                        pessoa.add(cli2);
                        pessoa.add(fun2);

                        for (int i = 0; i < pessoa.size(); i++) {
                                System.out.println(pessoa.get(i));
                                if (i == 1) {
                                        System.out.println();
                                }
                        }
                }
}

abstract class Pessoa {

                String nome;
                String cpf;
```

```
	Pessoa(String nome, String cpf) {
		this.nome = nome;
		this.cpf = cpf;
	}

	public String toString() {
		return "Nome: " + nome +
				" - CPF: " + cpf;
	}
}

class Funcionario extends Pessoa {

	int matricula;
	ArrayList<Atendimento> atendimento = new ArrayList<>();

	Funcionario(String nome, String cpf, int matricula, Atendimento atendimento) {
		super(nome, cpf);
		this.matricula = matricula;
		this.atendimento.add(atendimento);
	}

	public String toString() {

		String retorno = " - Matricula: " + matricula;

		for (int i = 0; i < atendimento.size(); i++) {
			retorno = retorno + "\nAtendimento " + i + ": " +
atendimento.get(i);
		}

		return "Funcionario: " + super.toString() + retorno;
```

```
	}

}

class Cliente extends Pessoa {

	String telefone;
	ArrayList<Aparelho> aparelho = new ArrayList<>();

	Cliente(String nome, String cpf, String telefone, Aparelho aparelho) {
		super(nome, cpf);
		this.telefone = telefone;
		this.aparelho.add(aparelho);
	}

	public String toString() {
		String retorno = " - Telefone: " + telefone;

		for (int i = 0; i < aparelho.size(); i++) {
			retorno = retorno + "\nAparelho: " + aparelho.get(i);
		}

		return "Cliente: " + super.toString() + retorno;
	}
}

class Atendimento {

	Aparelho aparelho;
	String data;
	String descricao;

	Atendimento(String data, String descricao, Aparelho aparelho) {
```

```
                this.data = data;
                this.descricao = descricao;
                this.aparelho = aparelho;
        }

        public String toString() {
                return "Data: " + data +
                        " - Descrição: " + descricao;
        }
}

class Aparelho {

        ArrayList<Atendimento> atendimento = new ArrayList<>();
        String codigo;
        int ano;
        String marca;
        String modelo;

        public void adicionaAtendimento(Atendimento atendimento) {
                this.atendimento.add(atendimento);
        }

        Aparelho(String codigo, int ano, String marca, String modelo) {
                this.codigo = codigo;
                this.ano = ano;
                this.marca = marca;
                this.modelo = modelo;
        }

        public String toString() {
                String atendi = "Codigo: " + codigo +
                        " - Ano: " + ano +
```

```
                                        " - Marca: " + marca +
                                        " - Modelo: " + modelo;

                        return atendi;
                }
}
```

Exercício 4:

```
public class CadastroDocumentos {
        public static void main(String[] args) {
                int numero;
                String remetente, tipo;

                FabricaDocumentos carta = new FabricaDocumentos();
                numero = 234;
                remetente = "Fabricio";
                tipo = "Cartas";
                carta.criaDocumento(numero, remetente, tipo);

                FabricaDocumentos telegrama = new FabricaDocumentos();
                numero = 521;
                remetente = "José";
                tipo = "Telegramas";
                carta.criaDocumento(numero, remetente, tipo);

                FabricaDocumentos notificacoes= new FabricaDocumentos();
                numero = 191;
                remetente = "Ana Júlia";
                tipo = "Notificacoes";
                carta.criaDocumento(numero, remetente, tipo);

        }
}
```

```
abstract class Documento {
		int numero;
		String remetente;
		String tipo;

		Documento(int numero, String remetente) {
			this.numero = numero;
			this.remetente = remetente;
		}
}

class Cartas extends Documento {
		Cartas(int numero, String remetente) {
			super(numero, remetente);

			System.out.format("A carta n°: %d - Remetente: %s\n", numero,
remetente);
		}

}

class Telegramas extends Documento {
		Telegramas(int numero, String remetente) {
			super(numero, remetente);
			System.out.format("O telegrama n°: %d - Remetente: %s\n", numero,
remetente);
		}
}

class Notificacoes extends Documento {
		Notificacoes(int numero, String remetente) {
			super(numero, remetente);
			System.out.format("A Notificacoes n°: %d - Remetente: %s\n", numero,
remetente);
```

```
	}
}

abstract class Fabrica {
	public abstract Documento criaDocumento(int numero, String remetente,String tipo);
}

class FabricaDocumentos extends Fabrica {

	public Documento criaDocumento(int numero, String remetente,String tipo) {
		if (tipo.equals("Cartas")) {
			return new Cartas(numero, remetente);
		} else if (tipo.equals("Telegramas")) {
			return new Telegramas(numero, remetente);
		} else if (tipo.equals("Notificacoes")) {
			return new Notificacoes(numero, remetente);
		} else {
			System.out.println("Tipo errado");
		}
	}
}
```