

Questão 1:

Herança em POO é a habilidade das classes se relacionarem e as descendentes herdam características e comportamentos da classe progenitora.

Questão 2:

herança simples: cada classe pode ter apenas uma superclasse, entretanto, uma superclasse pode ter várias descendentes.

herança múltipla: uma classe pode herdar características e comportamentos de mais de uma superclasse.

Questão 3:

```
package MHVI;

import java.util.Calendar;

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

        Calendar dataNascPessoa = Calendar.getInstance();
        dataNascPessoa.set(1995, Calendar.JANUARY, 07);
        Pessoa pessoa = new Pessoa("Claudinha", "123456789",
dataNascPessoa);

        Calendar dataNascAluno = Calendar.getInstance();
        dataNascAluno.set(2002, Calendar.MAY, 02);

        Pessoa aluno = new Aluno("Caillou", "1098765432",
dataNascAluno, "11111111");

        Calendar dataNascFuncionario = Calendar.getInstance();
        Calendar dataAdmissaoFuncionario = Calendar.getInstance();
        dataNascFuncionario.set(2000, Calendar.FEBRUARY, 07);
        dataAdmissaoFuncionario.set(2021, Calendar.MARCH, 10);

        Pessoa funcionario = new Funcionario("Roberval", "1234765432",
dataNascFuncionario, "88888888",
            dataAdmissaoFuncionario, 3396.76f);

        Calendar dataNascChef = Calendar.getInstance();
        Calendar dataAdmissaoChef = Calendar.getInstance();
        Calendar dataPromocaoChef = Calendar.getInstance();
        dataNascChef.set(1990, Calendar.SEPTEMBER, 10);
        dataAdmissaoChef.set(2019, Calendar.SEPTEMBER, 06);
        dataPromocaoChef.set(2021, Calendar.APRIL, 10);
```

```

        Pessoa chefdepartamento = new ChefeDepartamento("João",
"456789123", dataNascChef, "987456", dataAdmissaoChef,
        10000, "RH", dataPromocaoChef, 2000);

        System.out.println(pessoa.toString() + "\n" + aluno.toString()
+ "\n" + funcionario.toString() + "\n"
        + chefdepartamento.toString());
    }
}

```

```

package MHVI;

import java.util.Calendar;

public class Pessoa {

    protected String nome;
    protected String cpf;
    protected Calendar dataNasc;

    public Pessoa(String nome, String cpf, Calendar dataNasc) {
        this.nome = nome;
        this.cpf = cpf;
        this.dataNasc = dataNasc;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getCpf() {
        return cpf;
    }

    public void setCpf(String cpf) {
        this.cpf = cpf;
    }
}

```

```

        public Calendar getDataNasc() {
            return dataNasc;
        }

        public void setDataNasc(Calendar dataNasc) {
            this.dataNasc = dataNasc;
        }

        @Override
        public String toString() {
            return "Pessoa [cpf=" + cpf + ", dataNasc=" +
dataNasc.get(Calendar.DAY_OF_MONTH) + "/"
            + dataNasc.get(Calendar.MONTH) + "/" +
dataNasc.get(Calendar.YEAR) + "/" + ", nome=" + nome + "]";
        }
    }

package MHVI;

import java.util.Calendar;

public class Aluno extends Pessoa {
    private String matricula;

    public Aluno(String nome, String cpf, Calendar dataNasc, String
matricula) {
        super(nome, cpf, dataNasc);
        this.matricula = matricula;
        // TODO Auto-generated constructor stub
    }

    public String getMatricula() {
        return matricula;
    }

    public void setMatricula(String matricula) {
        this.matricula = matricula;
    }

    @Override
    public String toString() {

```

```

        return "Aluno [cpf=" + cpf + ", dataNasc=" +
dataNasc.get(Calendar.DAY_OF_MONTH) + "/"
        + dataNasc.get(Calendar.MONTH) + "/" +
dataNasc.get(Calendar.YEAR) + "/" + ", nome=" + nome
        + ", matricula=" + matricula + "]";
    }
}

```

```

package MHVI;

import java.util.Calendar;

public class Funcionario extends Pessoa {

    protected String matricula;
    protected Calendar dataAdmissao;
    protected float salario;

    public Funcionario(String nome, String cpf, Calendar dataNasc,
String matricula, Calendar dataAdmissao,
        float salario) {
        super(nome, cpf, dataNasc);
        this.matricula = matricula;
        this.salario = salario;
        this.dataAdmissao = dataAdmissao;
        // TODO Auto-generated constructor stub
    }

    public String getMatricula() {
        return matricula;
    }

    public void setMatricula(String matricula) {
        this.matricula = matricula;
    }

    public Calendar getDataAdmissao() {
        return dataAdmissao;
    }

    public void setDataAdmissao(Calendar dataAdmissao) {
        this.dataAdmissao = dataAdmissao;
    }
}

```

```

    }

    public float getSalario() {
        return salario;
    }

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

    @Override
    public String toString() {
        return "Funcionario [cpf=" + cpf + ", dataNasc=" +
dataNasc.get(Calendar.DAY_OF_MONTH) + "/" +
        + dataNasc.get(Calendar.MONTH) + "/" +
dataNasc.get(Calendar.YEAR) + "/" + ", nome=" + nome
        + ", dataAdmissao=" +
dataAdmissao.get(Calendar.DAY_OF_MONTH) + "/" +
dataAdmissao.get(Calendar.MONTH)
        + dataAdmissao.get(Calendar.YEAR) + "matricula=" +
matricula + ", salario=" + salario + "]\n";
    }
}

```

```

package MHVI;

import java.util.Calendar;

public class ChefeDepartamento extends Funcionario {

    private String departamento;
    private Calendar datapromocao;
    private float gratificacao;

    public ChefeDepartamento(String nome, String cpf, Calendar
dataNasc, String matricula, Calendar dataAdmissao,
        float salario, String departamento, Calendar datapromocao,
float gratificacao) {
        super(nome, cpf, dataNasc, matricula, dataAdmissao, salario);
        this.datapromocao = datapromocao;
        this.gratificacao = gratificacao;
        this.departamento = departamento;
    }
}

```

```
// TODO Auto-generated constructor stub
}

public String getDepartamento() {
    return departamento;
}

public void setDepartamento(String departamento) {
    this.departamento = departamento;
}

public Calendar getDatapromocao() {
    return datapromocao;
}

public void setDatapromocao(Calendar datapromocao) {
    this.datapromocao = datapromocao;
}

public float getGratificacao() {
    return gratificacao;
}

public void setGratificacao(float gratificacao) {
    this.gratificacao = gratificacao;
}

@Override
public String toString() {
    return "ChefeDepartamento [cpf=" + cpf + ", dataNasc=" +
dataNasc.get(Calendar.DAY_OF_MONTH) + "/"
        + dataNasc.get(Calendar.MONTH) + "/" +
dataNasc.get(Calendar.YEAR) + "/"
        + ", nome=" + nome + ", dataAdmissao=" +
dataAdmissao.get(Calendar.DAY_OF_MONTH) + "/"
        + dataAdmissao.get(Calendar.MONTH) +
        "/" + dataAdmissao.get(Calendar.YEAR) + "matricula=" +
matricula + ", salario=" + salario
        + ", dataPromocao=" +
datapromocao.get(Calendar.DAY_OF_MONTH) + "/" + ", dataPromocao="
        + datapromocao.get(Calendar.MONTH) +
        "/" + ", dataPromocao=" +
datapromocao.get(Calendar.YEAR) + ", departamento=" + departamento
```

```

        + ", gratificação" + gratificacao + "]];

    }

}

```

questão 4:

O polimorfismo vem da palavra polimorfo, essa característica permite que diferentes objetos respondam a uma mesma mensagem de formas diferentes.

Questão 5:

polimorfismo de sobreposição - cria uma sobreposição do método de classe mãe com o método da classe descendente.

polimorfismo de inclusão - usa a capacidade de substituição da herança de uma classe mãe por qualquer classe descendente, para permitir um comportamento polimórfico nos métodos que usam a classe mãe.

Questão 6:

```

package MHVI;

import java.util.ArrayList;
import java.util.List;

public class Questao6 {
    public List<Pessoa2> lista = new ArrayList<Pessoa2>();

    public void cadastrarPessoa(Pessoa2 pessoa) {
        lista.add(pessoa);
    }

    public void mostrarPessoa() {
        for (Pessoa2 pessoa : lista) {
            System.out.println(pessoa.toString());
        }
    }

    public static void main(String[] args) {

        Questao6 aplicacao = new Questao6();

        Pessoa2 pessoa = new Pessoa2("Roberval", "123456789");

        Cliente cliente = new Cliente("Marivaldo", "109876543", 332);
    }
}

```

```

        Funcionario2 funcionario = new Funcionario2("Fulaninho",
"567896543", "325457", 3500);

        Gerente gerente = new Gerente("Creitin", "965436789", "0004",
5000, 1);

        aplicacao.cadastrarPessoa(pessoa);
        aplicacao.cadastrarPessoa(cliente);
        aplicacao.cadastrarPessoa(funcionario);
        aplicacao.cadastrarPessoa(gerente);

        aplicacao.mostrarPessoa();
    }
}

```

```

package MHVI;

public class Pessoa2 {

    protected String nome;
    protected String cpf;

    public Pessoa2(){

    }

    public Pessoa2(String nome, String cpf) {
        this.nome = nome;
        this.cpf = cpf;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getCpf() {
        return cpf;
    }
}

```



```

    public void setCpf(String cpf) {
        this.cpf = cpf;
    }

    @Override
    public String toString() {
        return "Pessoa [cpf=" + cpf + ", nome=" + nome + "]";
    }
}

```

```

package MHVI;

public class Funcionario2 extends Pessoa2{

    protected String matricula;
    protected float salario;

    public Funcionario2(){

    }

    public Funcionario2(String nome, String cpf, String matricula,
float salario) {
        super(nome, cpf);
        this.matricula = matricula;
        this.salario = salario;
    }

    public String getMatricula() {
        return matricula;
    }

    public void setMatricula(String matricula) {
        this.matricula = matricula;
    }

    public float getSalario() {
        return salario;
    }

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

```

```

@Override
public String toString() {
    return "Funcionario [cpf=" + cpf + ", nome=" + nome + ",
matricula=" + matricula + ", salario=" + salario + "];"
}
}

```

```

package MHVI;

public class Cliente extends Pessoa2{

    protected int codigo;

    public Cliente(){

    }

    public Cliente(String nome, String cpf, int codigo) {
        super(nome, cpf);
        this.codigo = codigo;
    }

    public int getCodigo() {
        return codigo;
    }

    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }

    @Override
    public String toString() {
        return "Cliente [cpf=" + cpf + ", nome=" + nome + ", codigo=" +
codigo + "];"
    }
}

package MHVI;

public class Gerente extends Funcionario2 {

    protected int area;

```

```
public Gerente() {

}

    public Gerente(String nome, String cpf, String matricula, float
salario, int area) {
        super(nome, cpf, matricula, salario);
        this.area = area;
    }

    public int getArea() {
        return area;
    }

    public void setArea(int area) {
        this.area = area;
    }

    @Override
    public String toString() {
        return "Gerente [cpf=" + cpf + ", nome=" + nome + ",
matricula=" + matricula + ", salario=" + salario + ", area=" + area +
" ]";
    }

}
```