

Implementar as tarefas abaixo utilizando os conceitos de **Orientação a Objetos**:

1. Fatorial: <https://br.spoj.com/problems/FATORIA2/>
2. Posição i-ésima: <https://br.spoj.com/problems/QUERM/>
3. Cartas: <https://br.spoj.com/problems/CARTAS14/>
4. Carnaval: <https://br.spoj.com/problems/CARNA12/>
5. Fliper: <https://br.spoj.com/problems/JFLIPE14/>
6. Notas: <https://br.spoj.com/problems/NOTAS14/>
7. Língua do P: <https://br.spoj.com/problems/LINGUA14/>
8. Recursividade: <https://br.spoj.com/problems/F91/>
9. Atividade - Classe Imóvel e Interface Alugável

Crie uma interface chamada `Alugavel` que declare um método `calcularValorAluguel()`, que retorna um valor do tipo `double`.

Crie uma classe chamada `Imovel` que tenha um atributo `valorImovel` do tipo `double` e um método `calcularValor()`, que retorna o valor do imóvel.

Crie uma classe chamada `Apartamento` que herde da classe `Imovel` e implemente a interface `Alugavel`. Essa classe deve conter dois atributos privados: `valorImovel` e `valorAluguel`, ambos do tipo `double`.

Crie um construtor para inicializar esses atributos.

Crie um objeto da classe `Apartamento`, imprima os valores retornados ao chamar os métodos `calcularValor()` e `calcularValorAluguel()`.

10. Atividade - Pessoa e Funcionário

Crie uma classe chamada `Pessoa` com um construtor que receba os parâmetros `String nome` e `int idade` e inicialize os atributos correspondentes.

Crie um método `imprimirDados()` que exiba o nome e a idade da pessoa.

Crie uma classe chamada `Funcionario` que herde da classe `Pessoa` e que tenha um atributo privado `double salario`.

Crie um construtor para inicializar todos os atributos da classe `Funcionario`.

Sobrescreva o método `imprimirDados()` para incluir o salário do funcionário.

Crie um objeto da classe `Funcionario`, passe o nome, idade e salário como argumentos do construtor e chame o método `imprimirDados()`.

11. Atividade - Classe Abstrata e Herança

Crie uma classe abstrata chamada `Funcionario` que contenha os atributos `String nome` e `double salario`.

Adicione um método abstrato `imprimirDados()`.

Crie as classes `PessoaJuridica` e `PessoaFisica`, que herdam da classe `Funcionario`.

A classe `PessoaJuridica` deve conter os atributos `String cnpj` e `LocalDate dataCriacao`.

A classe `PessoaFisica` deve conter os atributos `String cpf` e `LocalDate dataNascimento`.

Sobrescreva o método `imprimirDados()` de cada classe para exibir os dados no seguinte formato:

`PessoaJuridica`: "Nome Fantasia: \$nome, CNPJ: \$cnpj, Data de criação: \$dataCriacao, Salário: \$salario"

`PessoaFisica`: "Nome: \$nome, CPF: \$cpf, Data de nascimento: \$dataNascimento, Salário: \$salario"

Crie um método static chamado `imprimirDados(Funcionario funcionario)` dentro da classe `Main`, que recebe um objeto `Funcionario` como parâmetro e chama seu método `imprimirDados()`.

Crie um objeto das classes `PessoaJuridica` e `PessoaFisica` e utilize o método `imprimirDados()` para exibir seus dados.

12. Atividade - Criando um Objeto de Funcionario

Utilizando a classe `Funcionario` da atividade anterior, crie um objeto e execute o método `imprimirDados()`.

Caso necessário, modifique a classe `Funcionario`, mas o que foi feito na atividade anterior deve continuar funcionando corretamente.