

**Programa de FORMAÇÃO DE TALENTOS PÚBLICA TECNOLOGIA**

Olá! Obrigado pelo interesse em querer ser mais um talento em nosso time.

O objetivo deste teste é verificar suas habilidades de codificação e arquitetura.

Para isso convidamos você, para responder a **um teste teórico e um teste prático** onde poderá mostrar suas técnicas de desenvolvimento.

**Teste Teórico**

1. Considere que o seguinte código de uma linguagem hipotética não contém erros de sintaxe para responder às questões 1 a 6.

***Importante: para responder a cada questão, selecione a alternativa correta no formulário.***

classe Pessoa{

constante Pessoa pai

constante Pessoa mae

variavel Lista<Pessoa> filhos

inicializadorPessoa(pai, mae, filhos){

self.pai = pai

self.mae = mae

self.filhos = filhos

}

}

classe Dados{

function getPessoasFromDB(){

variavel ListaDePessoas<Pessoa> = retornaTodasAsPessoasDoDB()

return ListaDePessoas

}

function retornaPais(){

variavel ListaDePessoas<Pessoa> = getPessoasFromDB()

variavel ListaDePais = nova Lista<Pessoa>

for pessoa in ListaDePessoas{

ListaDePais.acrescentar(pessoa.pai)

}

return ListaDePais

}

function retornaFilhos(){

variavel ListaDePessoas<Pessoa> = getPessoasFromDB()

variavel ListaDeFilhos = nova Lista<Pessoa>

for pessoa in ListaDePessoas{

for i = 1; i <  pessoa.filhos.tamanho; i++{

ListaDeFilhos.acrescentar(filhos[i])

}

}

return ListaDeFilhos

}

}

constante dadosDB = Dados()

variavel pais = DadosDB.retornaPais()

variavel filhos = DadosDB.retornaFilhos()

* 1. Sobre a variável filhos, é possível afirmar:

É uma lista de objetos do tipo Pessoa onde Pessoa.filhos nunca é nulo.

Não tem itens repetidos.

É uma lista de objetos do tipo Filho.

Contém todos os filhos das pessoas cadastradas no banco.

Contém alguns filhos das pessoas cadastradas no banco.

* 1. Sobre a variável pais, é correto afirmar que:

Nenhum item dessa lista é nulo.

Não tem itens repetidos.

É uma lista de objetos do tipo Pai.

Contém todos os pais cadastrados no banco.

Para cada item dessa lista, a propriedade filhos tem um tamanho maior que 1.

1.3. Considerando o padrão MVC, pode-se afirmar:

Pessoa é uma View.

Dados é um Modelo e Pessoa um Controller.

retornaPais() é um ViewModel.

Dados e Pessoa são Modelos.

Dados é Controller e Pessoa é Modelo.

1.4. Qual é o tipo de dadosDB

Constante

Variável

Dados

Pessoa

Nulo

1.5. O método getPessoasFromDB

Não é uma função.

Retorna uma Lista de um tipo específico.

Cria uma visualização.

É um exemplo de modelo.

Usa Pessoa como único parâmetro.

1.6. A instrução ListaDeFilhos.acrescentar(filhos[i])

Coloca na lista filhos o valor na posição i da ListaDeFilhos.

Causa um erro fatal se i for maior que o tamanho de ListaDeFilhos.

Sempre vai ter o valor i positivo.

Usa o método estático “acrescentar”.

Cria um loop infinito porque altera o objeto dentro do loop de iteração.

2. Considere que o seguinte código de uma linguagem hipotética não contém erro de sintaxe.

a = 0

if a >=0 {

print(a)

a++

}else if  a == 1{

print(a)

}

print(‘fim’)

Esse código imprimirá:

fim

0fim (resposta certa)

01fim

1

0

3. Sobre variáveis

Depois de definido o seu valor não muda.

Pode armazenar múltiplos valores simultaneamente.

Só podem ser usadas dentro do seu escopo.

Pode mudar de tipo em uma linguagem de tipagem forte.

Não podem ser reutilizadas.

4. Uma função

Pode ter apenas um parâmetro.

Retorna apenas um valor.

Sempre precisa ter um retorno.

Sempre precisa ter parâmetros.

Não pode usar variáveis.

5. Considerando que o seguinte código de uma linguagem hipotética não tem erros de sintaxe, quais são os valores retornados

i=0

while (i < 10){

print(i)

i++

if (i / 2 == 4) break

print(‘,’)

}

0,1,2,3,4,5,6,79,

0,1,2,3,5,6,79,

0,1,2,3,4,5,6,7

1,2,4,5,6,79,

0,1,2,3,4,5,6,79,10,

**Questão Prática:**

**Problema**

Maria é jogadora de basquete e criou uma tabela para registrar suas pontuações. A cada jogo ela anota quantos pontos conseguiu marcar e atualiza o mínimo e o máximo da temporada. Ela anota também nesta tabela quantas vezes quebrou o recorde mínimo e quantas vezes quebrou o recorde máximo. Os placares são sempre números inteiros e positivos menores que 1000 Depois de 4 jogos a tabela está assim:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Jogo | Placar | Mínimo da temporada | Máximo da temporada | Quebra recorde min. | Quebra recorde máx. |
| 1 | 12 | 12 | 12 | 0 | 0 |
| 2 | 24 | 12 | 24 | 0 | 1 |
| 3 | 10 | 10 | 24 | 1 | 1 |
| 4 | 24 | 10 | 24 | 1 | 1 |

Crie um programa na sua linguagem de programação preferida para facilitar o acompanhamento de resultados da Maria.

O objetivo deste teste é verificar suas habilidades de codificação e arquitetura.

Você PODE e DEVE utilizar bibliotecas e frameworks de terceiros conforme julgar necessário.

Nós encorajamos você a exagerar um pouco na solução para mostrar do que você é capaz, e com isso ganhar alguns ***“likes”***. Só não se esqueça de cumprir os requisitos mínimos

**Requisitos mínimos:**

* Possibilidade de inserção de novos jogos;
* Cálculo do mínimo e máximo da temporada;
* Cálculo da quantidade de vezes que o recorde foi quebrado;
* Interface para inserção dos dados;
* Interface para consulta dos dados;
* Testes unitários;
* Controle de versão Git;
* Documentação do código;

**Entrega da solução**

O seu código-fonte deve ser enviado para um repositório público no Github e o link desse repositório você deve enviar para o e-mail [publica@proway.com.br](mailto:publica@proway.com.br). Nesse e-mail você também irá anexar esse documento com as respostas assinaladas.

IMPORTANTE: No repositório do Github o arquivo README deve conter as orientações para rodar a sua aplicação.