|  |  |  |  |
| --- | --- | --- | --- |
| Aluno: | Vinicius Gustavo da Silva | RA: | 207035 |
| Aluno: | Josue Lara | RA: | 207455 |

Curso : GTI/ADS **Data:** 16/06/2020 

LEIA COM ATENÇÃO AS INSTRUÇÕES GERAIS

* Identifique corretamente esta prova.
* Valor da prova: 7 pontos

**QUESTÕES**

1. (0,5 ponto) Considere o seguinte diagrama de classe:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Conta Corrente** | | numero da conta  nome do correntista  saldo | | construtor (numero, nome, saldo)  alterarNome(nome)  deposito(valor)  saque(valor)  print() | | Considerando este diagrama responda:   1. Qual é o nome da classe?  |  | | --- | | O nome da classe é Conta Corrente. |  1. Quais são os atributos?  |  | | --- | | Os atributos são: | | numero da conta, nome do correntista e saldo. | |  | |  | |  |  1. Relacione os métodos da classe.  |  | | --- | | Os métodos são: | | constructor, alterarNome, deposito, saque e print. | |  | |  | |  | |

1. (2 pontos) **Classe Conta Corrente:** Crie uma classe para implementar uma conta corrente. A classe deve possuir os seguintes atributos: número da conta, nome do correntista e saldo.

Os métodos são os seguintes: alterarNome, depósito (só faz esta operação se o valor do depósito for maior que zero) , saque (verifica se possui saldo) e print; No construtor, saldo é opcional, com valor default zero e os demais atributos são obrigatórios.

|  |
| --- |
| **Conta Corrente** |
| numero da conta  nome do correntista  saldo |
| construtor (numero, nome, saldo)  alterarNome(nome)  deposito(valor)  saque(valor)  print() |

class ContaCorrente{

constructor (conta, nome, saldo){

this.conta = conta;

this.nome = nome;

this.saldo = saldo;

}

altera(nome){

this.nome = nome;

}

deposito(valor) {

if (valor > 0) {

this.saldo = this.saldo + valor;

} else{alert("Deposito precisa ser maior que zero!");}

}

saque(valor){

if (valor > 0) {

this.saldo = this.saldo - valor;

} else{alert("Saldo zerado");}

}

print(){

console.log("Conta: " + this.conta);

console.log("Cliente: " + this.nome);

console.log("Saldo = "+ this.saldo);

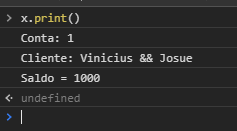
}

}

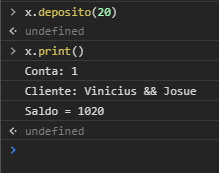
1. (0,5 ponto) Após a implementação da classe Conta Corrente, instancie um objeto e teste todos os métodos.

let x = new ContaCorrente(01, “Vinicius && Josué”, 1000)

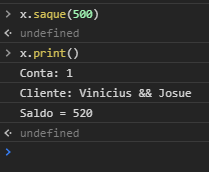
x.print()



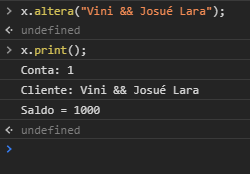
x.deposito(20)



x.saque(500)



x.altera(“Vini && Josué Lara”);



1. (1 ponto) Acrescente os métodos getter e setter.

set \_conta(conta){

this.conta = conta;

}

get \_conta(){

return this.conta;

}

set \_nome(nome){

this.nome = nome;

}

get \_nome(){

return this.nome;

}

set \_saldo(saldo){

this.saldo = saldo;

}

get \_saldo(){

return this.saldo;

}

1. (0,5 ponto) Uma dupla construiu o getter e setter para saldo da seguinte forma:

|  |  |
| --- | --- |
|  |  |

Não funcionou. Explique o motivo.

É necessário que haja o \_ ao definir os setters e getters para referenciar o objeto. Exemplo: se o código estivesse conforme abaixo, o mesmo funcionaria:

set \_saldo(saldo){

this.saldo = saldo;

}

get \_saldo(){

return this.saldo;

}

1. (2 pontos) Construa a Classe Fisica que é filha da classe conta corrente. A classe deve acrescentar o CPF. Os métodos são os seguintes: getter, setter e print e construtor. Lembrete: reutilize o que vem de herança.

class Fisica extends ContaCorrente{

constructor (conta, nome, saldo, cpf){

super (conta, nome, saldo);

this.cpf = cpf;

}

set \_cpf(cpf){

this.cpf = cpf;

}

get \_cpf(){

return this.cpf;

}

print (){

super.print();

console.log ("CPF: "+ this.cpf);

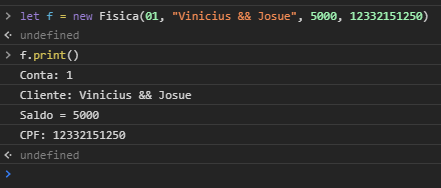
}

}

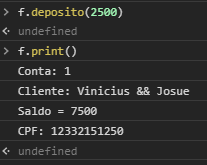
1. (0,5 ponto) Após a implementação da classe Física, instancie um objeto e teste todos os métodos.

let f = new Fisica(01, "Vinicius && Josue", 5000, 12332151250)

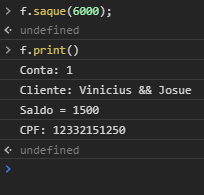
f.print();



f.deposito(2500);



f.saque(6000);



f.altera(“Vini && Josue Lara”);

