

Revisão da Prova I – Técnicas de programação I – Prof. Arley

1 – Marque a alternativa que descreve o que acontece quando executamos o comando `tsc ./index.ts`.

- a) Será criado o arquivo de configuração para compilação do tsc (TypeScript Compiler).
- b) O comando faz a verificação de tipos do arquivo `index.ts` e se houver erros de tipos, o TS os relatará no console, caso contrário o programa será executado.
- c) O comando se encarrega de gerar o arquivo de tipos `tsconfig.json` necessário para a execução do programa `index.ts`.
- d) O comando faz a compilação convertendo o código TS do arquivo `index.ts` em código JS puro e coloca no arquivo `index.js`.
- e) O comando faz a compilação e execução do código do arquivo `index.ts`.

2 – Marque a alternativa que descreve o papel do arquivo `package.json` em um projeto Node.js.

- a) É o arquivo de execução do projeto.
- b) É o arquivo de configuração do projeto.
- c) É o arquivo de compilação do projeto.
- d) É o arquivo de dependências do projeto.
- e) É o arquivo de scripts do projeto.

3 – Marque a alternativa que descreve o papel da pasta `node_modules` em um projeto Node.js.

- a) Local onde se encontra o código executável de um projeto baseado em Node.js.
- b) Local onde se encontra o código compilado e executável de um projeto baseado em Node.js.
- c) Local onde se encontra o código fonte do Node.js necessário para executar um projeto baseado em Node.js.
- d) Local onde se encontra o versionamento de um projeto baseado em Node.js.

- e) Local onde estão todas as dependências do projeto. As dependências diretas, instaladas usando `npm install`, e as dependências indiretas.

4 – Marque a alternativa que descreve o papel do arquivo `tsconfig.json` em um projeto Node.js.

- a) O arquivo de configuração `tsconfig.json` é usado para especificar as opções de compilação, incluindo a versão da ECMAScript para a qual o TS deve ser compilado.
- b) O arquivo de execução `tsconfig.json` é usado para especificar as opções de execução, incluindo a versão de JS compatível.
- c) O arquivo de compilação `tsconfig.json` é usado para especificar as opções de compilação, incluindo a versão de JS compatível.
- d) O arquivo de compilação `tsconfig.json` é usado para especificar as opções de compilação e execução, incluindo a versão de JS compatível.
- e) O arquivo de configuração `tsconfig.json` é usado para especificar as dependências do projeto.

5 – Marque a alternativa que descreve o motivo do código TS a seguir apresentar erro na atribuição da 2ª linha.

```
let entrada = "12";  
entrada = parseInt(entrada);
```

- a) Por atribuição explícita a variável `entrada` é declarada como `string`, impedindo de receber um valor `number`.
- b) Por tipagem estática a variável `entrada` é declarada como `string`, impedindo de receber um valor `number`.
- c) Por anotação explícita de tipo (`type annotation`), a variável `entrada` é declarada como `string`, impedindo de receber um valor `number`.

Revisão da Prova I – Técnicas de programação I – Prof. Arley

- d) Por inferência de tipo (type inference), o TS declara a variável entrada com o tipo string, impedindo de receber um valor number.
- e) Por tipagem dinâmica a variável entrada é declarada como string, impedindo de receber um valor number.

6 - Analise o trecho de código e marque a alternativa que possui uma instrução que apresenta o erro no código.

```
const nros = [11,22,33];
nros[2] = 0.5;
nros[3] = 1.5;
console.log(nros[2]);
console.log(nros[3]);
nros = [0.5,0.8,0.1];
```

- a) nros[2] = 0.5;
- b) nros[3] = 1.5;
- c) console.log(nros[2]);
- d) console.log(nros[3]);
- e) nros = [0.5,0.8,0.1];

7 – Os tipos genéricos são uma alternativa para definir a tipagem de variáveis. Marque a alternativa que possui uma atribuição correta na variável inputs.

```
const inputs:Array<string> = [];
```

- a) inputs[0] = 1;
- b) inputs = ["a","b"];
- c) inputs = [1,2,3];
- d) inputs = ["a",2];
- e) inputs = [true];

8 – A união de tipos é uma forma de combinar dois ou mais tipos em um único tipo, permitindo que uma variável possa aceitar diferentes tipos de valores. Marque a alternativa que possui uma atribuição **incorreta** na variável inputs.

```
let inputs:Array<string|number> = [];
```

- a) inputs[0] = 1;
- b) inputs = ["a","b"];
- c) inputs = [1,2,3];
- d) inputs = ["a",2];
- e) inputs = [true];

9 – A anotação de tipos pode usar a união de tipos na definição de arrays usando a notação de colchetes e a notação de tipos genéricos. Marque a alternativa que possui uma atribuição **incorreta** na variável.

- a) let a: string[] | number[] = ["a","b"];
- b) let b: string[] | number[] = ["a",2];
- c) let c: string[] | number[] = [1,2];
- d) let d: (string|number)[] = ["a",2];
- e) let e:Array<number|string> = ["a",2];

10 - No TS, é possível definir parâmetros condicionais em uma função utilizando tipos condicionais. O tipo condicional é definido colocando ? após a declaração da variável. Desta forma, a variável terá valor undefined se ela não receber valor como parâmetro. Marque a alternativa que possui uma chamada **incorreta** da função **exibir**.

```
function exibir(a:number, b?:number,
               c?:number, d?:number):void {
    console.log(a,b,c,d);
}
```

- a) exibir(1,2,3,4);
- b) exibir(1,2,3);
- c) exibir(1,2);
- d) exibir(1);
- e) exibir();

11 – As arrow functions são funções anônimas que possuem uma sintaxe compacta quando compara às funções anônimas e nomeadas. A notação compacta é interessante para ser utilizada como função inline (na

Revisão da Prova I – Técnicas de programação I – Prof. Arley

linha) em outras instruções. Marque a alternativa que possui uma declaração **incorreta** de arrow function.

- a) `const a = () => console.log("oi");`
- b) `const b = (x:number,y:number) => x + y;`
- c) `const c = (x:number) => { console.log(x); };`
- d) `const d = (x:number) => return x + 1;`
- e) `const e = (x:number,y:number) => { return x + y; };`

12 – Uma função passada para outra como parâmetro é chamada de função call-back (chamar de volta). Marque a alternativa que possui aquilo que será exibido no console.

```
const sum = (x:number,y:number) => x + y;
const dif = (x:number,y:number) => x - y;
const calc = (f:Function,a:number,b:number)
=> f(a,b);
console.log(calc(sum,1,2));
console.log(calc(dif,1,2));
```

- a) 3 e -1
- b) 3 e 3
- c) sum e dif
- d) calc, sum e dif
- e) Alguma mensagem de erro

13 – Uma função pode ser atribuída para uma variável. Marque a alternativa que possui aquilo que será exibido no console.

```
const inc = (x:number) => x + 1;
const calc = inc;
const res = calc(10);
console.log(calc);
console.log(res);
```

- a) 11 e 11
- b) Function e Function
- c) Function e 11
- d) 11 e Function
- e) Function e 10

14 – A POO (Programação Orientada a Objetos) permite a definição de tipos de dados através de classes. A classe é

uma estrutura que define um tipo de dado composto por operações (métodos) e propriedades (variáveis definidas no escopo da classe). Marque a alternativa que possui a instrução que constrói um objeto e chama o método print desse objeto.

```
class A {
    x:number;
    y:number;

    constructor(x:number,y:number) {
        this.x = x;
        this.y = y;
    }

    print() {
        console.log(this.x, this.y);
    }
}
```

- a) `const a = A(1,2); c.print();`
- b) `const b = A.print(3,4);`
- c) `const c = new A.print(5,6);`
- d) `const d = new A(7,8); d.print();`
- e) `const e = A(7,8).print();`

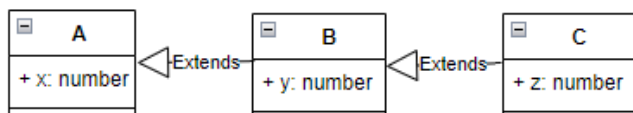
15 – Qual é o papel da palavra-reservada this no código do exercício anterior?

- a) A palavra-reservada this é usada para fazer referência à instância atual da classe.
- b) A palavra-reservada this é usada para fazer referência à classe atual.
- c) A palavra-reservada this é usada para fazer referência ao construtor da classe.
- d) A palavra-reservada this é usada para fazer referência à operação dentro da classe.
- e) A palavra-reservada this é usada para fazer referência às propriedades da classe.

16 – A herança é um recurso da POO que permite que uma classe herde propriedades e métodos de outra

Revisão da Prova I – Técnicas de programação I – Prof. Arley

classe. Isso permite criar uma hierarquia de classes, onde as subclasses herdam o comportamento e as características da superclasse. Marque a alternativa que possui um código **incorreto** considerando o diagrama UML de classes a seguir.



- a) class C extends B {
 z:number;
 constructor(x:number, y:number, z:number){
 super(x,y);
 this.z = z;
 }
}
- b) class A {
 x:number;
 constructor(x:number){
 this.x = x;
 }
}
- c) class B extends A{
 y:number;
 constructor(x:number, y:number){
 this.y = y;
 }
}
- d) const b = new B(1,2);
- e) const c = new C(1,2,3);

17 – Qual é o papel da palavra-reservada super na herança?

- a) Ela é usada para criar a ligação entre a superclasse (classe pai) e a subclasse (classe filha).
- b) Ela é usada para referenciar a instância da superclasse (classe pai).
- c) Ela é usada para construir uma instância da superclasse (classe pai).
- d) Ela é usada para criar a operação de herança entre as classes.

e) Ela é usada para chamar métodos e construtores da classe pai (superclasse) a partir da classe filha (subclasse).

18 – Analise o trecho de código a seguir e marque a alternativa correta.

```

class X {
    nome:string = "Ana";

    print(){
        console.log(this.nome);
    }
}

class Y extends X {
    nome:string = "Maria";

    print(){
        console.log(this.nome);
    }
}
  
```

- a) O método print foi sobrescrito na classe Y.
- b) A propriedade nome foi sobrescrita na classe Y.
- c) A propriedade nome foi sobrecarregada na classe Y.
- d) O método print foi sobrecarregado na classe Y.
- e) A classe X foi sobrescrita na classe Y.

19 – Analise o trecho de código a seguir e marque a alternativa que imprime no console o texto Boa tarde.

```

class X {
    msg() {
        console.log("Bom dia");
    }
}

class Y extends X {}

class Z extends X {
    msg() {
        console.log("Boa tarde");
    }
}
  
```

- a) const a:X = new X(); a.msg();

Revisão da Prova I – Técnicas de programação I – Prof. Arley

- b) `const b:Y = new Y(); b.msg();`
- c) `const c:X = new Y(); c.msg();`
- d) `const d = new Y(); d.msg();`
- e) `const e:X = new Z(); e.msg();`

20 – O termo polimorfismo é originário do grego e significa muitas formas. Na POO o polimorfismo permite que classes mais abstratas representem o comportamento de classes concretas. Marque a alternativa correta com relação ao polimorfismo.

- a) No código do Exercício 16 fez se uso de polimorfismo.
- b) No código do Exercício 18 fez se uso de polimorfismo.
- c) No código do Exercício 19 fez se uso de polimorfismo.
- d) No código dos Exercício 16 e 18 fez se uso de polimorfismo.
- e) No código dos Exercício 18 e 19 fez se uso de polimorfismo.