FATEC JAHU

AVALIAÇÃO P1 – ESTRUTURA DE DADOS – 10/10/2024

Prof. Me. Tiago Antonio da Silva

INSTRUÇÕES

I. Leiam a prova com atenção e discutam as soluções em voz baixa.
II. A prova é em dupla e com consulta apenas entre o material da dupla.
III. Ao término da prova, envie seus arquivos JavaScript com o nome da dupla comentado no código.
IV. A prova se inicia às 9h30m e terminará às 13h00m. Não serão aceitos envios tardios. Tempo mínimo de permanência: 1h30m.
V. COMENTEM TODO O CÓDIGO EXPLICANDO O FUNCIONAMENTO DE CADA FUNCIONALIDADE.
VI. Dúvidas? Retorne ao primeiro item.

QUESTÕES

1) Analise o seguinte trecho de código e identifique possíveis erros ou melhorias. Explique suas observações.

```javascript
class Pilha {
    constructor() {
        this.itens = [];
    }

    push(elemento) {
        this.itens[this.itens.length] = elemento;
    }

    pop() {
        if (this.itens.length == 0) {
            return "Pilha vazia";
        }
        let topo = this.itens[this.itens.length - 1];
        delete this.itens[this.itens.length - 1];
        return topo;
    }

    peek() {
        return this.itens[this.itens.length];
    }
}

let pilha = new Pilha();
pilha.push(1);
pilha.push(2);
console.log(pilha.pop());  // Esperado: 2
console.log(pilha.peek()); // Esperado: 1
```

2) Implementem uma fila de atendimento de um hospital em JavaScript sem usar funções nativas de arrays. Nesta fila, pacientes com maior gravidade têm prioridade sobre os pacientes menos graves, mesmo que tenham chegado depois.

Requisitos:

a) Adicionar Paciente: A função **`adicionarPaciente(nome, gravidade)`** deve adicionar um paciente à fila. A gravidade é representada por um número, onde um número maior indica maior gravidade.

b) Atender Paciente: A função **`atenderPaciente()`** deve remover e retornar o paciente com a maior gravidade presente na fila.

c) Listar Pacientes: A função **`listarPacientes()`** deve exibir todos os pacientes na ordem de atendimento, do mais grave para o menos grave.

d) Verificar Fila Vazia: A função **`estaVazia()`** deve retornar **`true`** se a fila estiver vazia, caso contrário, **`false`**.

3) O código a seguir pretende implementar uma fila, mas apresenta problemas. Identifique e corrija os erros para que a fila funcione corretamente.

```javascript
class Fila {
    constructor() {
        this.itens = {};
        this.inicio = 1;
        this.fim = 1;
    }

    enqueue(elemento) {
        this.itens[this.fim] = elemento;
        this.fim++;
    }

    dequeue() {
        if (this.isEmpty()) {
            return "Fila vazia";
        }
        let primeiro = this.itens[this.inicio];
        delete this.itens[this.inicio];
        this.inicio++;
        return primeiro;
    }

    isEmpty() {
        return this.fim === this.inicio;
    }

    peek() {
        return this.itens[this.inicio];
    }
}

let fila = new Fila();
fila.enqueue("A");
fila.enqueue("B");
fila.enqueue("C");
console.log(fila.dequeue()); // Esperado: "A"
console.log(fila.peek());    // Esperado: "B"
```