

```
export class CNPJ {
```

```
    private static readonly tamanhoCNPJSemDV: number = 12;
    private static readonly regexCNPJSemDV: RegExp = /^[A-Z\d]{12}$/;
    private static readonly regexCNPJ: RegExp = /^[A-Z\d]{12}(\d){2}$/;
    private static readonly regexCaracteresMascara: RegExp = /[.\/-]/g;
    private static readonly regexCaracteresNaoPermitidos: RegExp = /^[A-Z\d.\/-]/i;
    private static readonly valorBase: number = "0".charCodeAt(0);
    private static readonly pesosDV: number[] = [6, 5, 4, 3, 2, 9, 8, 7, 6, 5, 4, 3, 2];
    private static readonly cnpjZerado: string = "00000000000000";
```

```
    static isValid(cnpj: string): boolean {
        if (!this.regexCaracteresNaoPermitidos.test(cnpj)) {
            let cnpjSemMascara = this.removeMascaraCNPJ(cnpj);
            if (this.regexCNPJ.test(cnpjSemMascara) && cnpjSemMascara !== CNPJ.cnpjZerado) {
                const dvInformado = cnpjSemMascara.substring(this.tamanhoCNPJSemDV);
                const dvCalculado = this.calculaDV(cnpjSemMascara.substring(0, this.tamanhoCNPJSemDV));
                return dvInformado === dvCalculado;
            }
        }
        return false;
    }
}
```

```
    static calculaDV(cnpj: string): string {
        if (!this.regexCaracteresNaoPermitidos.test(cnpj)) {
            let cnpjSemMascara = this.removeMascaraCNPJ(cnpj);
            if (this.regexCNPJSemDV.test(cnpjSemMascara) && cnpjSemMascara !== this.cnpjZerado.substring(0, this.tamanhoCNPJSemDV)) {
                let somatorioDV1 = 0;
                let somatorioDV2 = 0;
                for (let i = 0; i < this.tamanhoCNPJSemDV; i++) {
                    const asciiDigito = cnpjSemMascara.charCodeAt(i) - this.valorBase;
                    somatorioDV1 += asciiDigito * this.pesosDV[i + 1];
                    somatorioDV2 += asciiDigito * this.pesosDV[i];
                }
                const dv1 = somatorioDV1 % 11 < 2 ? 0 : 11 - (somatorioDV1 % 11);
                somatorioDV2 += dv1 * this.pesosDV[this.tamanhoCNPJSemDV];
                const dv2 = somatorioDV2 % 11 < 2 ? 0 : 11 - (somatorioDV2 % 11);
                return `${dv1}${dv2}`;
            }
        }
        throw new Error("Não é possível calcular o DV pois o CNPJ fornecido é inválido");
    }
}
```

```
    private static removeMascaraCNPJ(cnpj: string): string {
        return cnpj.replace(this.regexCaracteresMascara, "");
    }
}
```

```
}
```