	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ **Curso: ADS** **Disciplina: Programação Orientada a Objetos Professor: Ely**

Exercício 02

1. Qual a diferença entre tipagem dinâmica e tipagem estática?

Tipagem estática é quando após a variável ter sido declarada, não é mais possível alterar o seu tipo, a tipagem dinâmica é a capacidade que a linguagem tem de definir o tipo da variável de forma automática de acordo com o valor dela.

2. Qual o principal problema do uso de tipagem dinâmica?

o principal problema da tipagem dinâmica é a falta de verificação de tipo em tempo de compilação, o que pode levar a erros em tempo de execução

3. Pesquise um exemplo na internet em que a tipagem dinâmica pode ser

problemático.

Uma função somar números em python, aceita dois argumentos `a` e `b`, e tenta somá-los. No entanto, devido à tipagem dinâmica do Python, não há verificação de tipo em tempo de compilação. Isso significa que você pode chamar a função com argumentos de tipos diferentes e potencialmente obter um resultado inesperado ou um erro em tempo de execução

4. Pesquise e exemplifique com um exemplo porque dizemos que a linguagem C, mesmo tendo tipagem estática, possui tipagem fraca.

Dizemos que a linguagem C possui tipagem fraca, por que ela permite que valores de tipos diferentes interajam entre si, como por exemplo, ela permite somar dois números, sendo um de valor inteiro e outro de valor flutuante.

5. Pesquise e, se encontrar, um exemplo onde o tipo any seria benéfico

O tipo ‘any’, é uma característica do TS que permite que a variável seja de qualquer tipo, um caso onde ele seria benéfico poderia ser em um caso de consulta de múltiplos dados, onde não seria necessário especificar o tipo deles, tornando a consulta mais facilitada.

6. Poderíamos dizer que a tipagem do TypeScript é fraca por uma variável do tipo number aceitar tanto inteiros como ponto flutuante?

Não, pois ao declarar que uma variável é do tipo number, você afirma que ela pertence tanto a classe inteiro quanto a ponto flutuante, sendo verificado em tempo de compilação, evitando qualquer tipo de erro.

7. Reescreva o exemplo abaixo, mantendo a quebra de linhas usando template strings e os valores Ely, 120.56 e TypeScript venham de variáveis declaradas separadamente e “interpoladas” na string:

```
Ely
My payment time is 120.56
and
my preffered language is TypeScript
```

RESPOSTA:https://www.typescriptlang.org/play?#code/DYUwLgBAdg9gti
CAuCBnMAnAllA5hAXggHIBRYAT2IChRIAHAQ10YSjBhSgFc4AjEBkIQAjACYADADo
ArADZa4CMBy5uzEHBTpseYcTAV6IVAGNs9MDWqmYUVDFBTgMXAAoABgBIA3rAQAvt
QAshQQTBRskGBYCBBYqBC+TCxRMEGMUAAm1HBh9BggAGZFgiBZyplqGvGJACpGIAD
K5liWHgCUQA

8. Configure o seu arquivo de configuração do TypeScript estudando a documentação (https://www.typescriptlang.org/tsconfig) com as seguintes opções: a. Alterar o local em que os arquivos *.js são gerados para a pasta uma pasta que não seja a mesma de onde estão os arquivos *.ts. Por exemplo uma pasta chamada build. Pesquise por outDir;
    b. allowUnreachableCode com valor true;
        https://www.typescriptlang.org/pt/tsconfig#allowUnreachableCode
    c. noImplicitAny com valor true
        https://www.typescriptlang.org/pt/tsconfig#noImplicitAny
    d. target com o valor ES3. Além disso, utilize a classe do exercício anterior e veja como ela é transpilada para JS;
    e. strictNullChecks para false;
    f. Crie exemplos que mostrem o efeito das alterações no arquivo de configuração.