Nome: Claudia Nunes

2.1. Efetue uma definição, através de enumeração, de todos os membros da sua família, considerando apenas parentesco direto de pais, irmãos e filhos.

Membros da família $\stackrel{def}{=}$ 1-Darly,2-Vera,3-Claudia,4-Debora,5-Adriana,

2.2. Efetue uma definição, através de enumeração, de cursos existentes na instituição de ensino em que você está estudando.

1- Meio Ambiente e Recursos Hídricos
2- Geoprocessamento
3- Desenvolvimento de Software Multiplataforma

2.3. Efetue uma definição, através de enumeração, de números primos. Um número é dito primo se só possui como divisor ele mesmo e o número um.

Número primo $\stackrel{def}{=}$ 2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,...

2.4. Efetue uma definição ostensiva, de um colega de classe.

Fernando Ferreira $\stackrel{def}{=}$ 

2.5. Efetue uma definição recursiva de uma sequência numérica cujos elementos são obtidos multiplicando-se, a partir do segundo elemento, o elemento anterior por 3. Considere que o primeiro elemento vale 2.

Primeiro termo da sequência numérica $\stackrel{def}{=}$2
Sequência numérica $\stackrel{def}{=}$ {2,18, 54,162,486, ...}

2.6. Utilizando linguagem de programação defina:

a) Uma enumeração para os meses do ano.

```
enum MesesDoAno {
Janeiro = 1,
```

```
Fevereiro =2,
Março = 3,
Abril = 4,
Maio = 5,
Junho,
Julho,
Agosto = 8,
Setembro,
Outubro,
Novembro,
Dezembro,
}
const mesAtual: MesesDoAno = MesesDoAno.Agosto;
console.log(`Estamos no mês:  ${mesAtual}-
${MesesDoAno[mesAtual]}`);
```

b) Uma enumeração para os dias da semana.

```
enum DiaDaSemanas {
  Domingo = 1,
  Segunda,
```

```
  Terça,
  Quarta,
  Quinta,
  Sexta,
}

const diaatual: DiaDaSemanas = DiaDaSemanas.Quarta;
console.log(`Estamos no dia da
Semana:  ${diaatual}-${DiaDaSemanas[diaatual]}-
feira`);
```

c) Uma função recursiva para o cálculo do fatorial de um número.
Exemplo: 1*2*3*4*5*6 = 720

```
function calcularFatorial(numero: number): number {

    if (numero === 0) {
        return 1;
    }
    return numero * calcularFatorial(numero - 1);
}

const numero = 6;
const fatorial = calcularFatorial(numero);
console.log(`O fatorial de ${numero} é
${fatorial}`);
```

d) Uma definição que corresponda a definição do tipo gênero-diferença para um uma pessoa que estude em uma faculdade. Utilize uma linguagem que dê suporte a herança.

```
class Pessoa {
    nome: string;
    idade: number;
```

```
    genero: string;

    constructor(nome: string, idade: number,
genero: string) {
      this.nome = nome;
      this.idade = idade;
      this.genero = genero;
    }

    apresentar() {
      console.log(`Olá, eu sou ${this.nome} e tenho
${this.idade} anos.`);
    }
  }

    class Estudante extends Pessoa {
    faculdade: string;
    curso: string;

    constructor(nome: string, idade: number,
genero: string, faculdade: string, curso: string) {
      super(nome, idade, genero);
      this.faculdade = faculdade;
      this.curso = curso;
    }

    apresentar() {
      super.apresentar();
      console.log(`Estou estudando na
${this.faculdade} no curso de ${this.curso}.`);
    }
  }
```

```
  const estudante1 = new Estudante("Cladia", 35,
"Feminino", "FATEC Jacareí", "Desenvolvimento de
Software");
  estudante1.apresentar();
```

2.7. Um veículo possui a capacidade de se mover, expressa pela alteração na sua coordenada de longitude e latitude. Um veículo elétrico é um veículo que possui como fonte de energia primária a eletricidade (armazenada em uma bateria). Um veículo elétrico e voador é um veículo que também possui a capacidade de se mover na vertical, expressa pela alteração de sua altitude em relação ao solo. Represente um veículo elétrico e voador utilizando uma cadeia
de herança. Defina o código-fonte representativo do modelo em um arquivo separado daquele que faz uso desse e, adicionalmente exemplifique o acesso e a modificação desses atributos através de chamada de suas operações.

```
class Veiculo {
    private coordenadaLongitude: number;
    private coordenadaLatitude: number;

    constructor(longitude: number, latitude:
number) {
      this.coordenadaLongitude = longitude;
      this.coordenadaLatitude = latitude;
    }

    mover(longitude: number, latitude: number) {
      this.coordenadaLongitude = longitude;
      this.coordenadaLatitude = latitude;
    }
  }
  export{Veiculo}
```

```
import { Veiculo } from "./Veiculo";

class VeiculoEletrico extends Veiculo {
  private bateria: number;
```

```
  constructor(longitude: number, latitude: number,
capacidadeBateria: number) {
    super(longitude, latitude);
    this.bateria = capacidadeBateria;
  }

  carregarBateria(novoNivel: number) {
    this.bateria = novoNivel;
  }
}

export{Veiculo,VeiculoEletrico,}
```

```
import { VeiculoEletrico } from
"./VeiculoEletrico";

class VeiculoEletricoVoador extends VeiculoEletrico
{
  private altitude: number;

  constructor(longitude: number, latitude: number,
capacidadeBateria: number) {
    super(longitude, latitude, capacidadeBateria);
    this.altitude = 0;
  }

  mudarAltitude(novaAltitude: number) {
    this.altitude = novaAltitude;
  }
}

export{VeiculoEletricoVoador}
```