Here are five code examples demonstrating the usage of classes in TypeScript:

1. Basic Class Definition:

```typescript
class Person {
    name: string;
    age: number;

    constructor(name: string, age: number) {
        this.name = name;
        this.age = age;
    }

    greet() {
        console.log(`Hello, my name is ${this.name} and I'm ${this.age} years old.`);
    }
}

const person = new Person('John', 30);
person.greet(); // Output: Hello, my name is John and I'm 30 years old.
```

2. Inheritance:

```typescript
class Animal {
    name: string;

    constructor(name: string) {
        this.name = name;
    }

    makeSound() {
        console.log('Animal makes a sound');
    }
}

class Dog extends Animal {
    constructor(name: string) {
        super(name);
    }

    makeSound() {
        console.log('Dog barks');
    }
}

const dog = new Dog('Buddy');
dog.makeSound(); // Output: Dog barks
```

3. Access Modifiers:

```typescript
class Car {
    private speed: number;

    constructor(speed: number) {
        this.speed = speed;
    }

    accelerate() {
        this.speed += 10;
    }

    getSpeed() {
        return this.speed;
    }
}

const car = new Car(50);
car.accelerate();
console.log(car.getSpeed()); // Output: 60
```

4. Static Methods:

```typescript
class MathUtils {
    static PI: number = 3.14159;

    static calculateArea(radius: number): number {
        return MathUtils.PI * radius * radius;
    }
}

const radius = 5;
const area = MathUtils.calculateArea(radius);
console.log('Area of the circle:', area); // Output: 78.53975
```

5. Readonly Properties:

```typescript
class Circle {
    readonly radius: number;

    constructor(radius: number) {
        this.radius = radius;
    }

    calculateArea(): number {
        return Math.PI * this.radius * this.radius;
    }
}

const circle = new Circle(7);
console.log('Radius:', circle.radius);
console.log('Area:', circle.calculateArea()); // Output: Radius: 7, Area: 153.93804002589985
```

These examples demonstrate various features of classes in TypeScript, including class definition, inheritance, access modifiers, static methods, and readonly properties. Classes in TypeScript provide a powerful and flexible way to structure and organize your code, making it easier to work with object-oriented programming concepts.