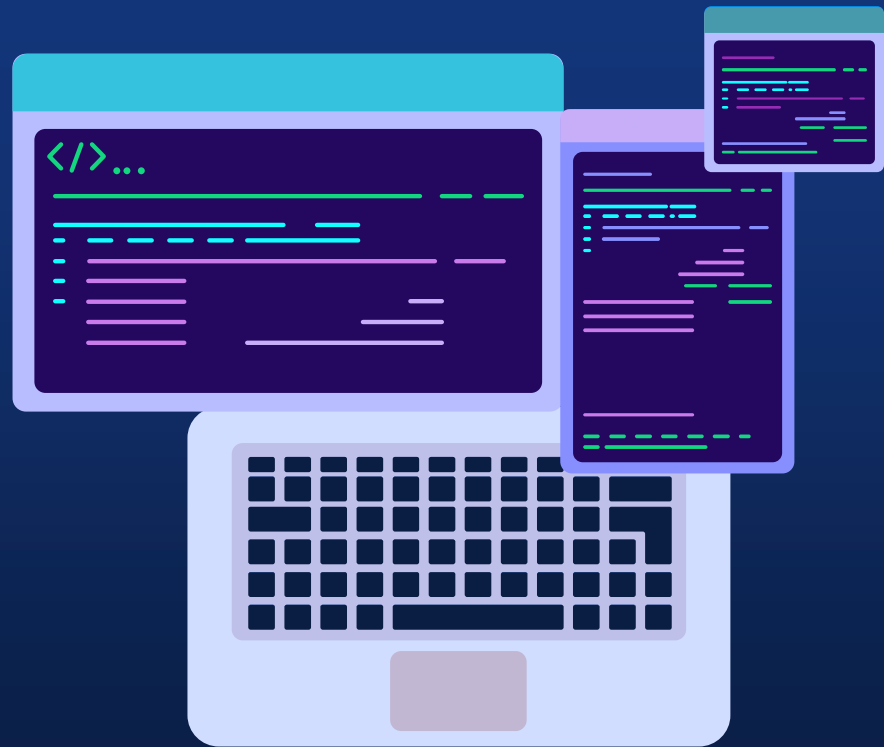


# Software Design and Important concepts



Mentor: Einar Rocha

# CONTENT

01

## OOP Pillars

Inheritance, Polymorphism  
Encapsulation, Abstraction

02

## Clean Code

Meaningful Names,  
Functions, Unit test  
Code Smells...

03

## SOLID

Single Responsibility  
Open closed  
Liskov Substitution  
Interface Segregation  
Dependency Inversion

04

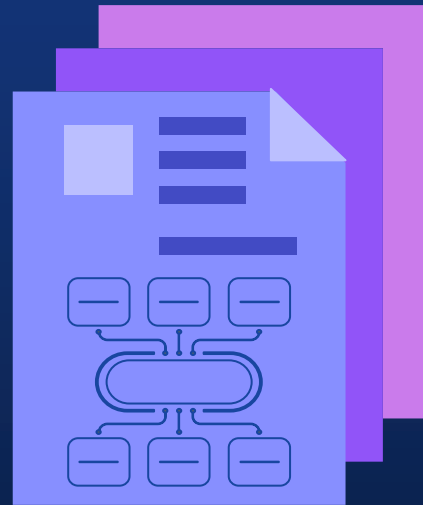
## Design patterns

Singleton, Factory Method  
Strategy, Observer  
Builder...



# 01

## Pillars of Object Oriented Programming (OOP)



# The Goals of Software Design



To allow us to write software that is as helpful as possible.



To allow our software to continue to be as helpful as possible.



To design systems that can be created and maintained as easily as possible by their programmers



# Agenda

## Inheritance

Class, Interface, Abstract Class,  
Virtual, Extension, Default

## Polymorphism

Dynamic Polymorphism,  
Static Polymorphism



```
var brand1 = "Sony";  
Console.WriteLine("The brand is: " + brand1);  
  
var brand2 = "Samsung";  
Console.WriteLine("The brand is: " + brand2);  
  
var brand3 = "Xiaomi";  
Console.WriteLine("The brand is: " + brand3);
```




## Procedural paradigm

```
static void Main(string[] args)
{
    printBrand("Sony");

    printBrand("Samsung");

    printBrand("Xiaomi");
}

private static void printBrand(string brand)
{
    Console.WriteLine("The brand is: " + brand);
}
```



Phone
- brand
- ShowBrand()

# What is OOP?

is a programming paradigm which allows us to solve problems via an object or more collection of collaborating objects



## OOP paradigm

```
class Phone
{
    private string brand;

    public Phone(string brand)
    {
        this.brand = brand;
    }

    public void ShowBrand()
    {
        Console.WriteLine(string.Format(
            "The brand is: {0}",
            this.brand)
        );
    }
}
```

**Inheritance**

**Polymorphism**

**Encapsulation and**

**Abstraction**

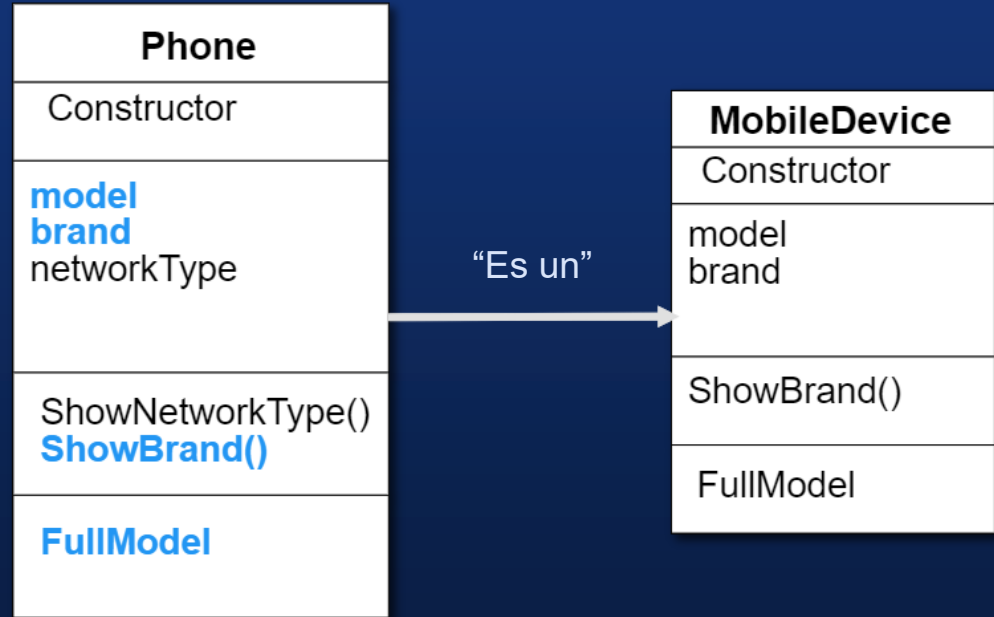


**new**

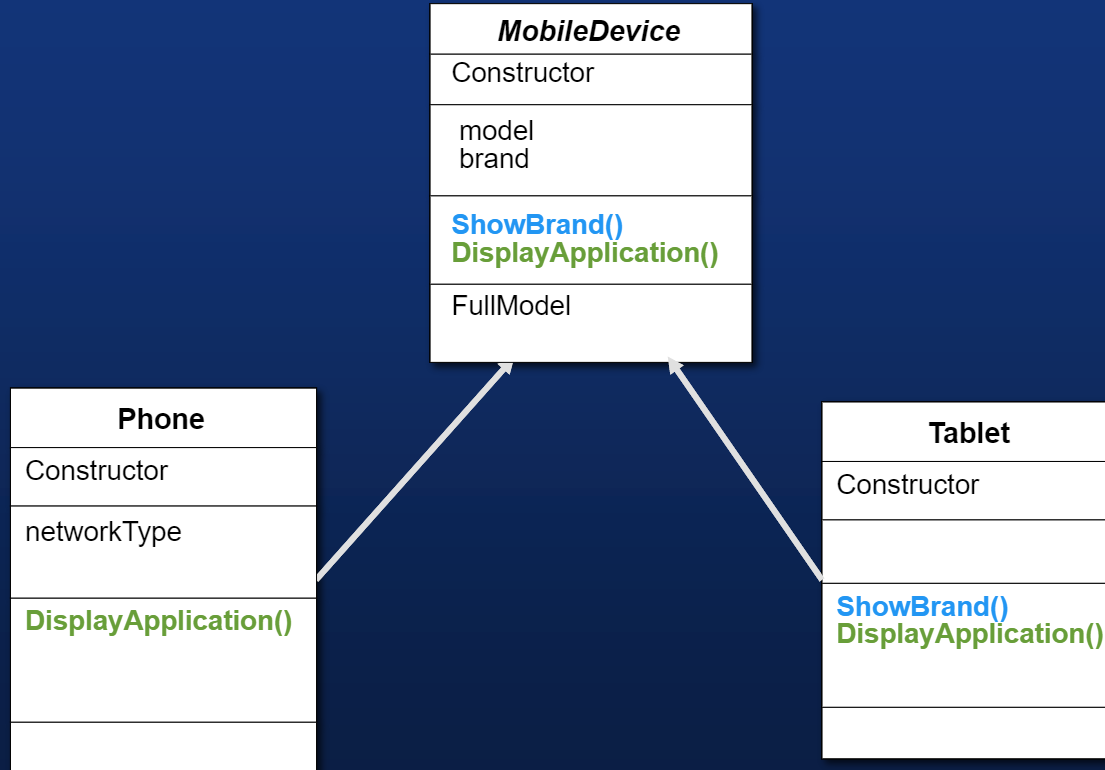


Class
Constructor
- serial - brand
- ShowBrand()
- FullModel

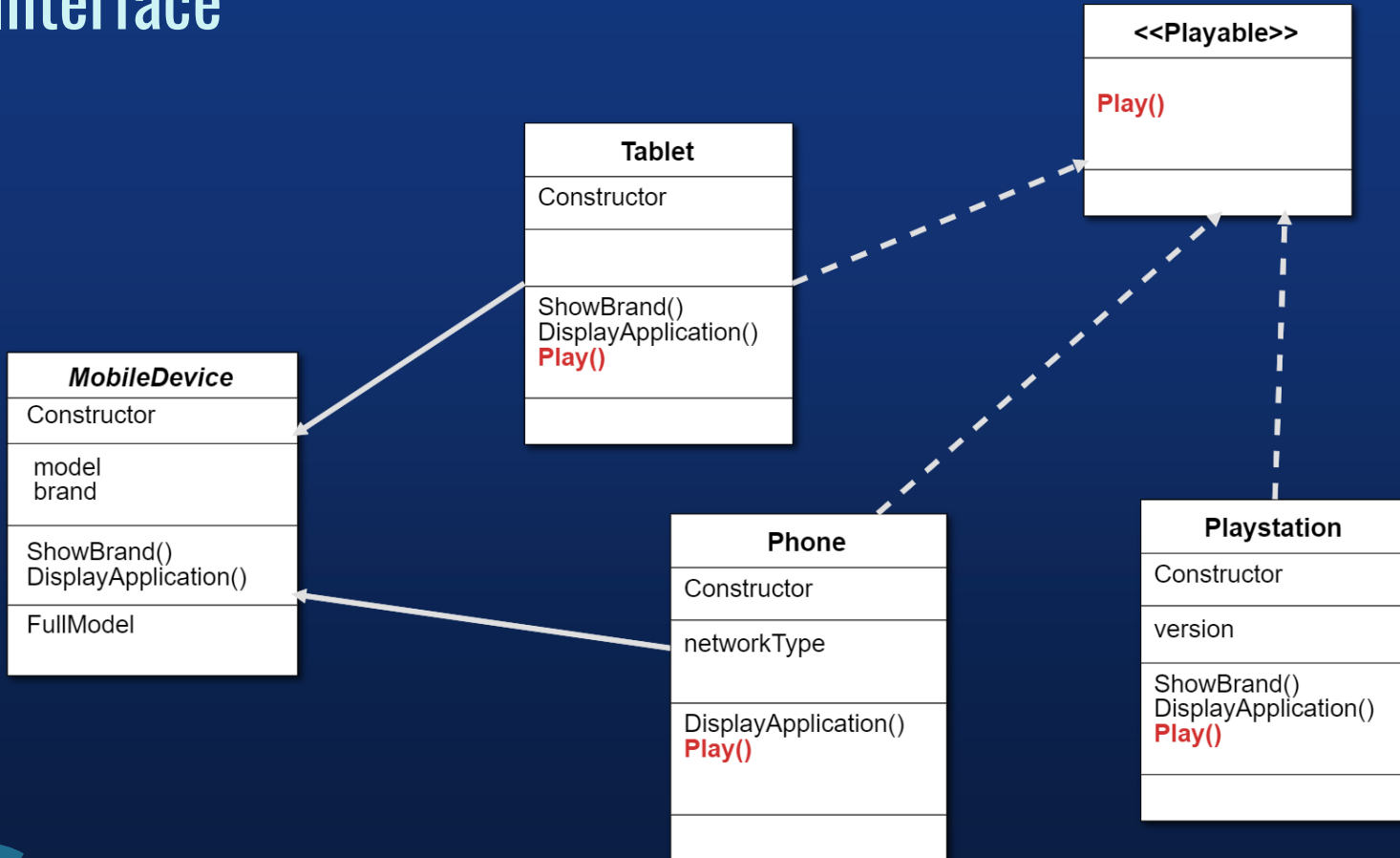
# Inheritance



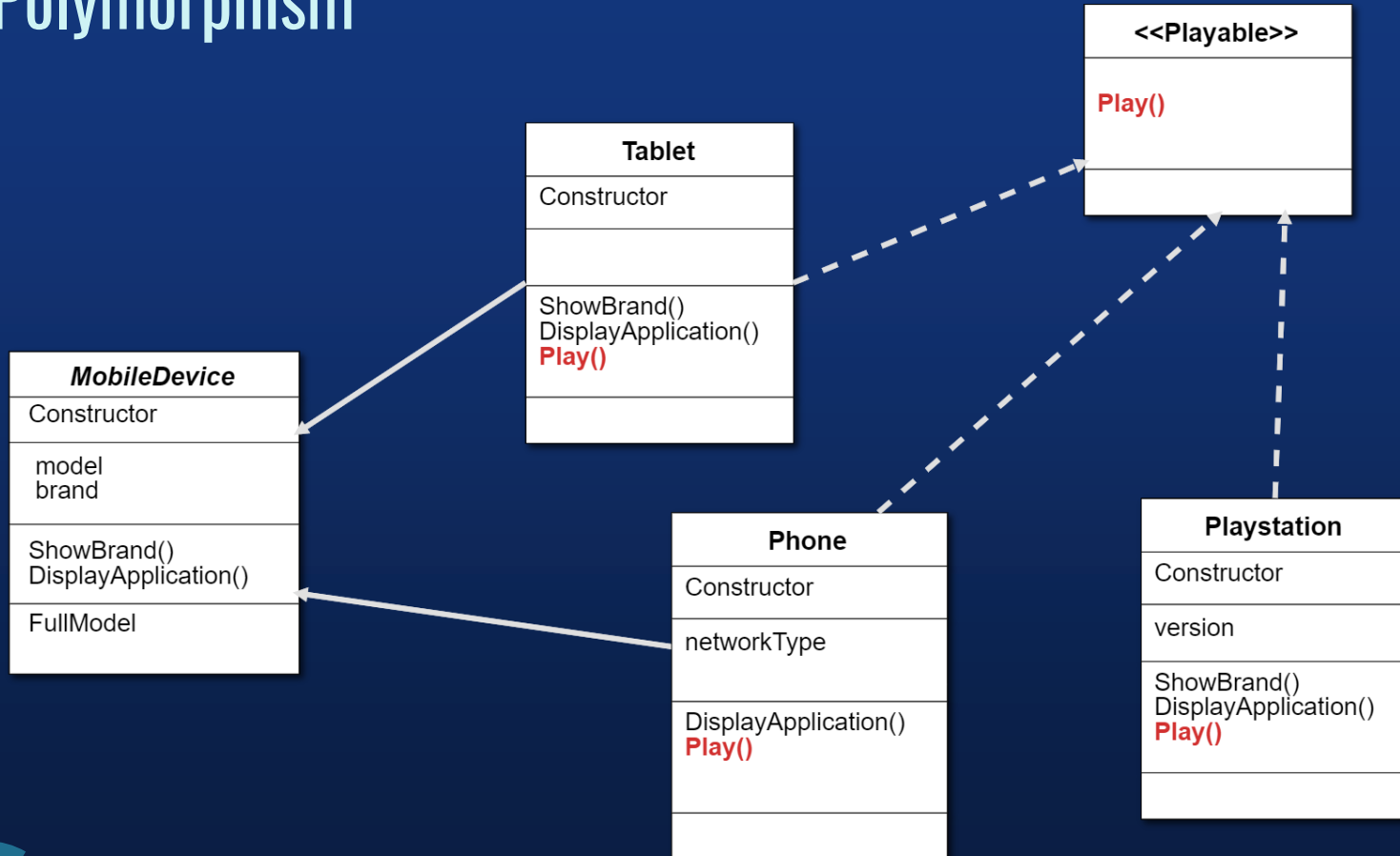
# Abstract Class, virtual keyword



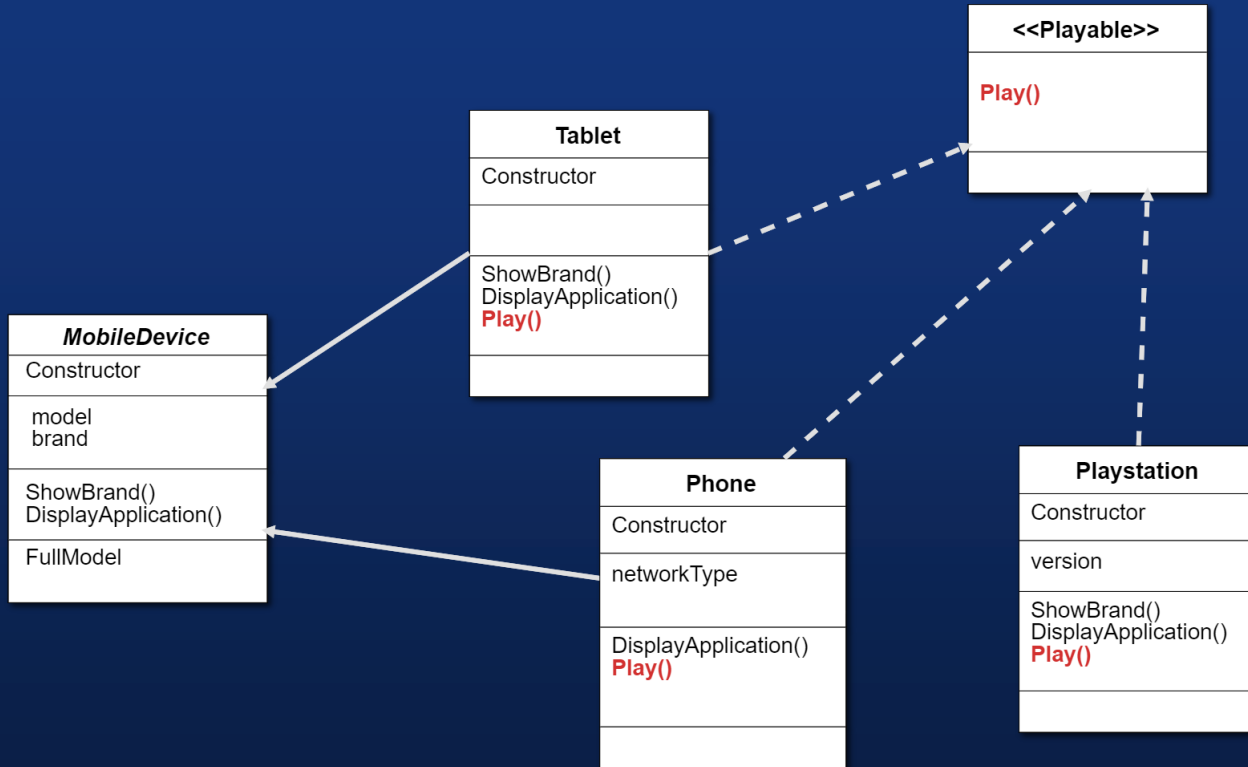
# Interface



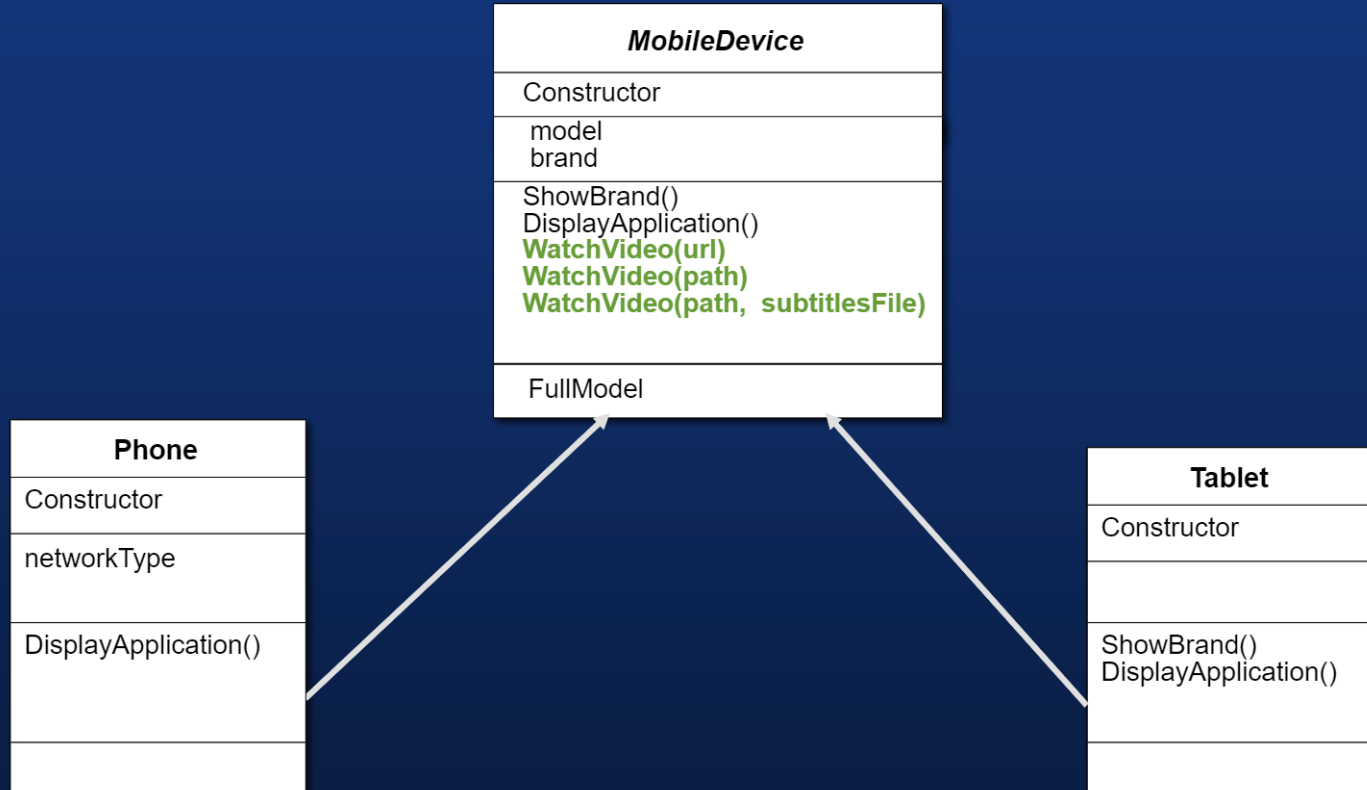
# Polymorphism



# Polymorphism

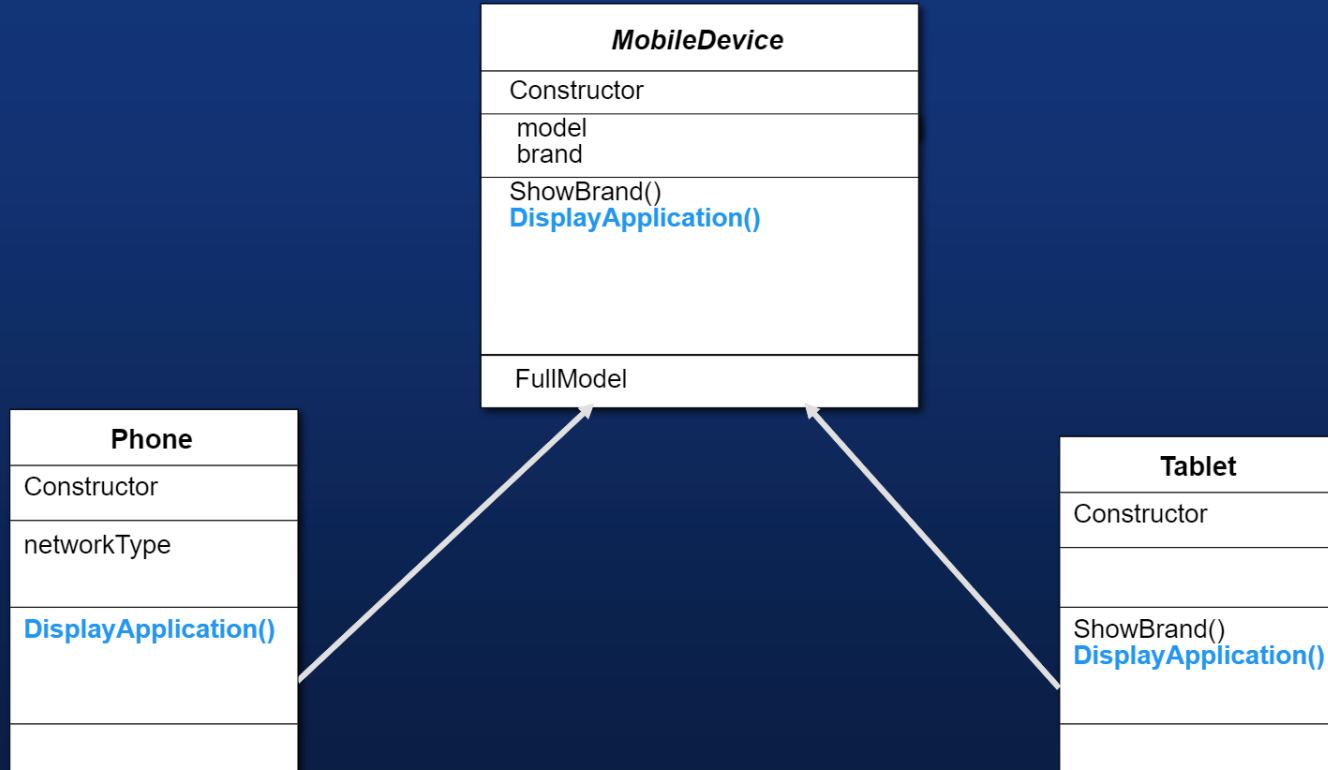


# Static Polymorphism





# Dynamic Polymorphism using Abstract class





# Summary

## Inheritance

- Inheritance is an OOP feature that allows one class to acquire attributes and behaviors from another class.
- The keyword “abstract” simply means that it does not have its own definition. And it must be defined in the inherited class.
- Virtual means it has a definition. However, it can be overwritten if you choose.
- Interfaces do not have constructors or fields. may have "Default interface methods in C # 8"
- Sealed classes are used to restrict the inheritance feature of OOP.

## Polymorphism

- Static polymorphism is achieved by method overloading
- Dynamic polymorphism is achieved by method overriding

