using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp2

{

abstract class Builder

{

public Computer computer = new Computer();

public abstract void CreateRam();

public abstract void CreateHardDisk();

public abstract void CreateProcessor();

public abstract void CreateKeyboard();

public abstract void CreateMonitor();

}

class AcerBuilder : Builder

{

public override void CreateHardDisk()

{

computer.HardDisk = "Acer Hard Disc";

}

public override void CreateKeyboard()

{

computer.Keyboard = "Acer Keyboard";

}

public override void CreateMonitor()

{

computer.Monitor = "Acer Monitor";

}

public override void CreateProcessor()

{

computer.Processor = "Acer Processor";

}

public override void CreateRam()

{

computer.Ram = "Acer Ram";

}

}

class AsusBuilder : Builder

{

public override void CreateHardDisk()

{

computer.HardDisk = "Asus Hard Disc";

}

public override void CreateKeyboard()

{

computer.Keyboard = "Asus Keyboard";

}

public override void CreateMonitor()

{

computer.Monitor = "Asus Monitor";

}

public override void CreateProcessor()

{

computer.Processor = "Asus Processor";

}

public override void CreateRam()

{

computer.Ram = "Asus Ram";

}

}

class Computer

{

public string Ram { get; set; }

public string HardDisk { get; set; }

public string Processor { get; set; }

public string Keyboard { get; set; }

public string Monitor { get; set; }

public override string ToString()

{

Console.WriteLine("=========COMPUTER=========");

Console.WriteLine($"Ram {Ram}");

Console.WriteLine($"Harddisk {HardDisk}");

Console.WriteLine($"Processor {Processor}");

Console.WriteLine($"Keyboard {Keyboard}");

Console.WriteLine($"Monitor {Monitor}");

Console.WriteLine("==========================");

return "";

}

}

class Director

{

public string Type { get; set; }

public void MakeType(string type)

{

Type = type;

}

public Computer MakeComputer(Builder builder)

{

if (Type == "simple")

{

builder.CreateHardDisk();

builder.CreateProcessor();

builder.CreateRam();

}

else if (Type == "premium")

{

builder.CreateHardDisk();

builder.CreateProcessor();

builder.CreateRam();

builder.CreateMonitor();

builder.CreateKeyboard();

}

return builder.computer;

}

}

class Program

{

static void Main(string[] args)

{

Director director = new Director();

director.MakeType("premium");

var computer = director.MakeComputer(new AsusBuilder());

Console.WriteLine(computer);

}

}

}