using System;

using System.Collections.Generic;

using System.Text;

namespace BridgePattern

{

public interface IMessageSender

{

void SendMessage(string Message);

}

public class SmsMessageSender : IMessageSender

{

public void SendMessage(string Message)

{

Console.WriteLine("'" + Message + "' : This Message has been sent using SMS");

}

}

public class EmailMessageSender : IMessageSender

{

public void SendMessage(string Message)

{

Console.WriteLine("'" + Message + "' : This Message has been sent using Email");

}

}

public abstract class AbstractMessage

{

protected IMessageSender messageSender;

public abstract void SendMessage(string Message);

}

public class ShortMessage : AbstractMessage

{

public ShortMessage(IMessageSender messageSender)

{

//Initialize the super class messageSender variable

this.messageSender = messageSender;

}

public override void SendMessage(string Message)

{

if (Message.Length <= 10)

{

messageSender.SendMessage(Message);

}

else

{

Console.WriteLine("Unable to send the message as length > 10 characters");

}

}

}

public class LongMessage : AbstractMessage

{

public LongMessage(IMessageSender messageSendor)

{

this.messageSender = messageSendor;

}

public override void SendMessage(string Message)

{

messageSender.SendMessage(Message);

}

}

class BrigdePattern2

{

static void Main(string[] args)

{

Console.WriteLine("Select the Message Type 1. For longmessage or 2. For shortmessage");

int MessageType = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Please enter the message that you want to send");

string Message = Console.ReadLine();

if (MessageType == 1)

{

AbstractMessage longMessage = new LongMessage(new EmailMessageSender());

longMessage.SendMessage(Message);

}

else

{

AbstractMessage shortMessage = new ShortMessage(new SmsMessageSender());

shortMessage.SendMessage(Message);

}

Console.ReadKey();

}

}

}