

# МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

# МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

# МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

# МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

**Факультет управления и информатики в технологических системах**

**Кафедра**[**корпоративных информационных систем и программирования**](https://vsuet.ru/obuchenie/faculties/uits/k_kisip)

**Направление подготовки 09.03.02**

***(шифр)***

**Информационные системы и технологии**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(наименование направления подготовки)*

**Отчет**

**по практическому заданию № 3**

Выполнил студент гр. У-213

Боровкова Анастасия Дмитриевна

(ф.и.о)

**\_\_\_\_\_\_\_\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***(****подпись****)***

Проверил(а):

Демичев С.Е.

**\_\_\_\_\_\_\_\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***(****подпись****)***

Воронеж – 2023

/\*\*

\* Приложение сцены в режиме моста

\*/

public class C01\_InScene {

public static void main(String[] args) {

// Создать конкретный объект реализации

MsgImplementor implementor = new SendBySMS() ;

// Создать обычный объект сообщения

AbstractMsg abstractMessage = new UserMsg(implementor);

abstractMessage.sendMessage («Вход в вашу учетную запись в разных местах», «Пользователь A0001»);

// Переключение в почтовый режим и ускорение обработки

implementor = new SendByEmail() ;

abstractMessage = new AdminMsg(implementor);

abstractMessage.sendMessage («Уведомление о запуске проекта», «Операция S0001»);

}

}

/\*\*

\* Инкапсулированный тип сообщения

\*/

abstract class AbstractMsg {

// Содержит объект, который реализует часть

MsgImplementor impl ;

public AbstractMsg (MsgImplementor impl){

this.impl = impl ;

}

/\*\*

\* Отправить сообщение и делегировать метод реализации

\* @param message - содержание сообщения для отправки

\* @param toUser - получатель сообщения

\*/

public void sendMessage (String message, String toUser){

this.impl.send(message, toUser);

}

}

class AdminMsg extends AbstractMsg{

public AdminMsg(MsgImplementor impl) {

super(impl);

}

@Override

public void sendMessage(String message, String toUser) {

message = "Крутой администратор:" + message;

super.sendMessage(message, toUser);

}

}

class UserMsg extends AbstractMsg{

public UserMsg(MsgImplementor impl) {

super(impl);

}

@Override

public void sendMessage(String message, String toUser) {

message = "Уважаемый пользователь:" + message;

super.sendMessage(message, toUser);

}

}

/\*\*

\* Инкапсулированная отправка сообщения

\*/

interface MsgImplementor {

void send (String message , String toUser) ;

}

class SendBySMS implements MsgImplementor{

@Override

public void send(String message, String toUser) {

System.out.println («SMS-уведомление:« + toUser + »; содержимое:« + сообщение »);

}

}

class SendByEmail implements MsgImplementor{

@Override

public void send(String message, String toUser) {

System.out.println ("Почтовое уведомление:" + toUser + "; Содержимое:" + сообщение);

}

}