абстракция

public abstract class AbstractPhone {

private int year;

public AbstractPhone(int year) {

this.year = year;

}

public abstract void call(int outputNumber);

public abstract void ring (int inputNumber);

}

инкапсуляция

public class SomePhone {

private int year;

private String company;

public SomePhone(int year, String company) {

this.year = year;

this.company = company;

}

private void openConnection(){

//findComutator

//openNewConnection...

}

наследование

public abstract class WirelessPhone extends AbstractPhone {

private int hour;

public WirelessPhone(int year, int hour) {

super(year);

this.hour = hour;

}

}

public class CellPhone extends WirelessPhone {

public CellPhone(int year, int hour) {

super(year, hour);

}

@Override

public void call(int outputNumber) {

System.out.println("Вызываю номер " + outputNumber);

}

@Override

public void ring(int inputNumber) {

System.out.println("Вам звонит абонент " + inputNumber);

}

}

полиморфизм

public abstract class AbstractPhone {

private int year;

public AbstractPhone(int year) {

this.year = year;

}

public printA(int a) {

system.out.print(a);

}

public printA(string a) {

system.out.print(a);

}

}