**Поверхностное и** **Глубокое** клонирование

В этом примере клонируется объект класса *Car*. Клонирование выполняется поверхностное - новый объект *clonedCar* содержит ссылку на тот же объект класса *Driver,* что и объект *car*. Если вас это не устраивает, то необходимо самим написать "глубокое" клонирование - создать новый объект класса *Driver*.

public class Car implements Cloneable {

private String name;

private Driver driver;

public Car(String name, Driver driver) {

this.name = name;

this.driver = driver;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Driver getDriver() {

return driver;

}

public void setDriver(Driver driver) {

this.driver = driver;

}

@Override

public Car clone() throws CloneNotSupportedException {

return (Car) super.clone();

}

}

public class Driver implements Cloneable {

private String name;

private int age;

public Driver(String name, int age) {

this.name = name;

this.age = age;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

@Override

public Driver clone() throws CloneNotSupportedException {

return (Driver) super.clone();

}

}

public class CloneCarDemo {

public static void main(String[] args) throws CloneNotSupportedException {

Car car = new Car("Грузовик", new Driver("Василий", 45));

Car clonedCar = car.clone();

System.out.println("Оригинал:\t" + car);

System.out.println("Клон: \t" + clonedCar);

Driver clonedCarDriver = clonedCar.getDriver();

clonedCarDriver.setName("Вася");

System.out.println("Оригинал после изменения имени водителя:\t" + car);

System.out.println("Клон после изменения имени водителя: \t\t" + clonedCar);

}

}

#### **Глубокое клонирование**

@Override

public Car clone() throws CloneNotSupportedException {

Car newCar = (Car) super.clone();

Driver driver = this.getDriver().clone();

newCar.setDriver(driver);

return newCar;

}