Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации»

Нижегородский институт управления

Кафедра Информатики и информационных технологий

ОТЧЕТ

Задания лекции №4

Выполнила: студент группы:

ИБ-321

Кодоева Нели Тамазиевна

Нижний Новгород

2023 г.

Оглавление

[**Задание 9.1** 3](#_Toc130758934)

[**Пункт 1** 3](#_Toc130758935)

[**Пункт 2** 5](#_Toc130758936)

[**Пункт 3** 7](#_Toc130758937)

[**Пункт 4** 8](#_Toc130758938)

[**Пункт 5** 9](#_Toc130758939)

[**Задание 9.2** 10](#_Toc130758940)

[**Пункт 1** 10](#_Toc130758941)

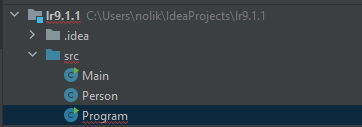
[**Пункт 2** 11](#_Toc130758942)

[**Пункт 3** 12](#_Toc130758943)

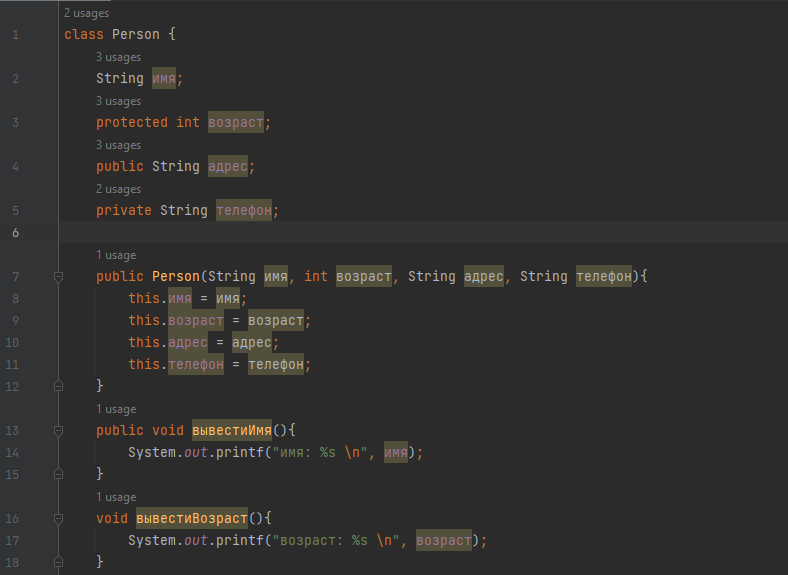
# **Задание 9.1**

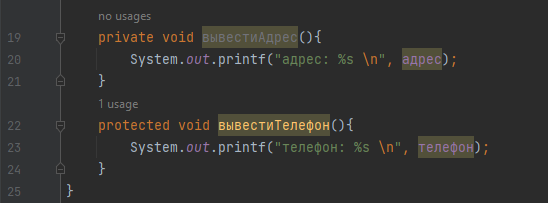
## **Пункт 1**

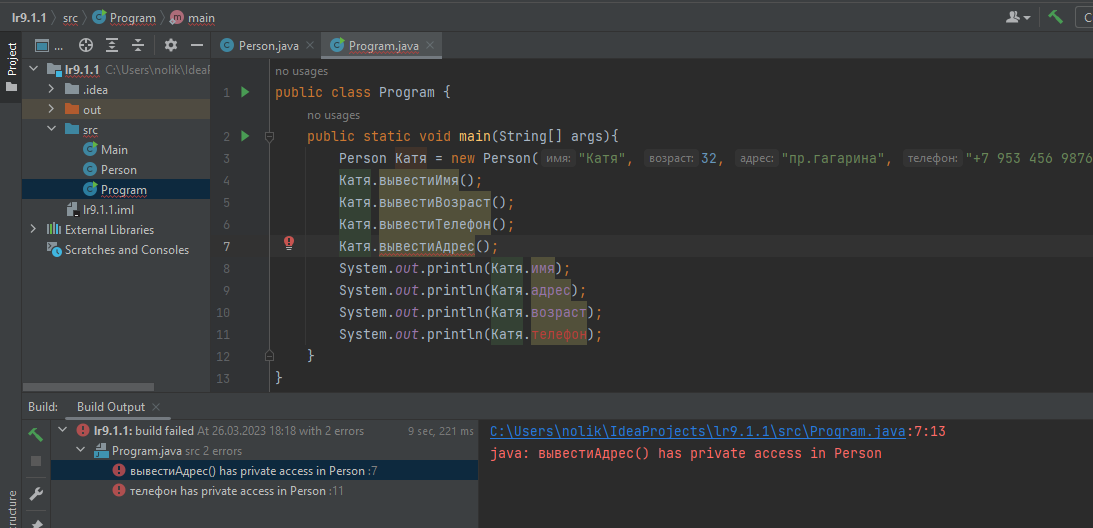
* 1. **Реализовать проекты с программами рисунков 9.1 и 9.2.**



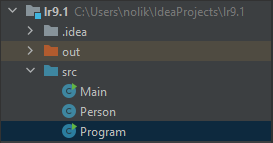




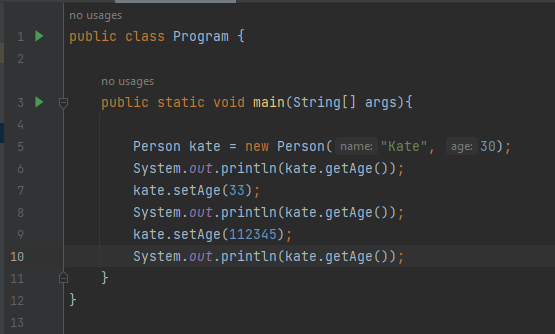


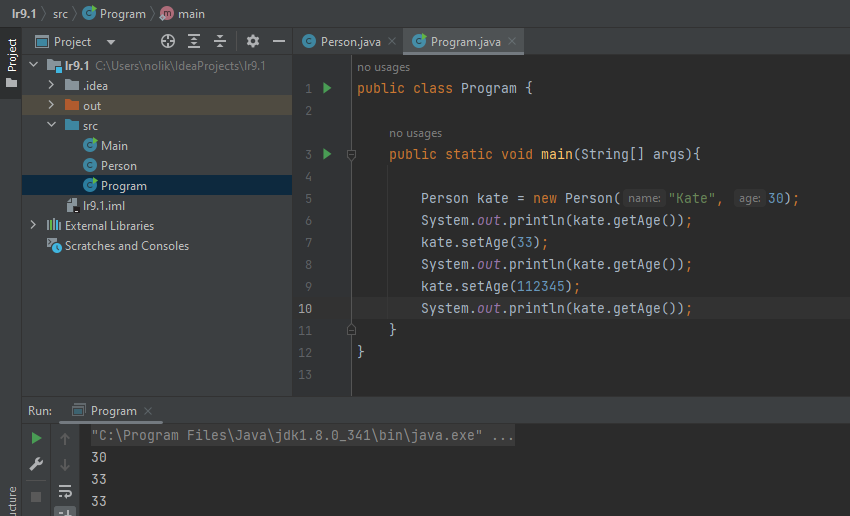


*Рис.1-5(реализация проекта с рисунка 9.1)*





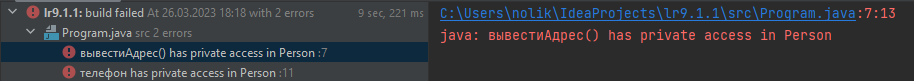


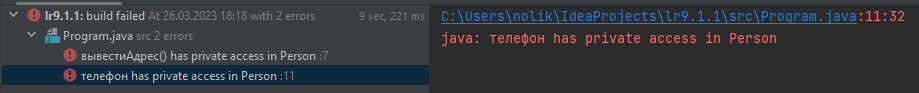


*Рис.6-9(реализация проекта с рисунка 9.2)*

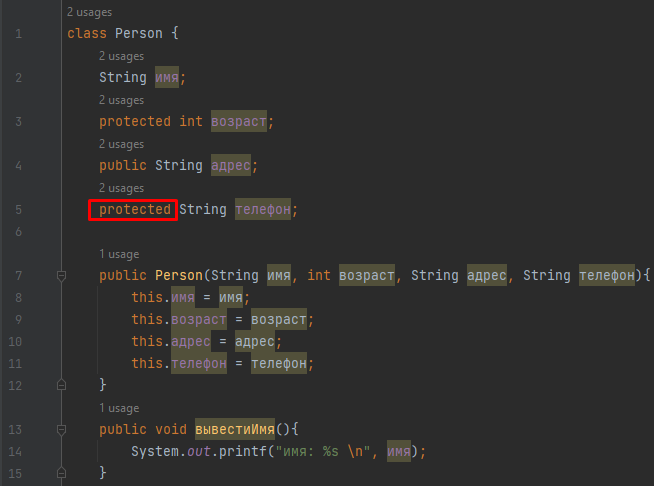
## **Пункт 2**

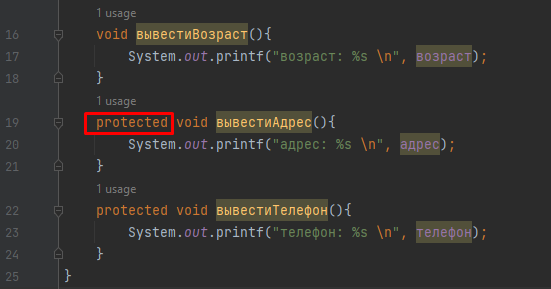
* 1. **Определить (идентифицировать) причины ошибок проекта рисунка 9.1 и исправить их.**

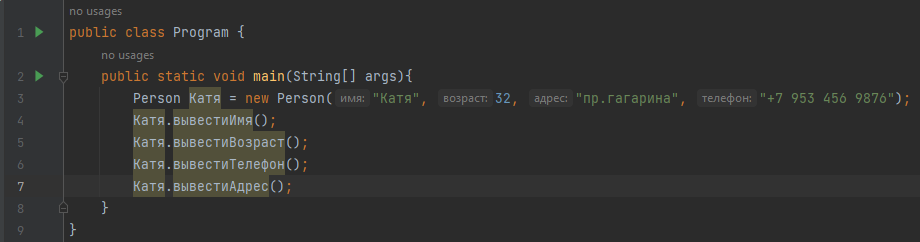


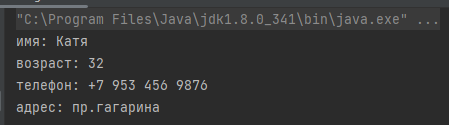


*Рис.10-11(строка ошибок при реализации проекта с рисунка 9.1)*









*Рис.12-15(исправленный проект и его реализация)*

**Пояснение:** в строке 5 и 19 класса Person модификаторы private были заменены на protected, так как поля и методы с модификатором private в классе не имеют доступа. Также в классе Program были убраны строки 8-11, потому что выведение результата на экран уже описано в классе Person.

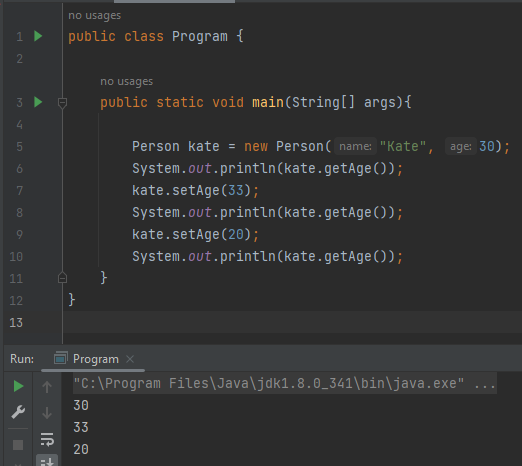
## **Пункт 3**

**1.3 В проекте рисунка 9.2 объяснить работу кодов строчек 6-10 и добиться вывода другого значения возраста по методу .setAge.**



*Рис.16(строки кода класса Person с рисунка 9.2)*

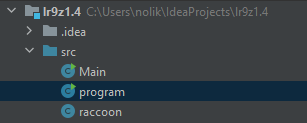
**Пояснение:** на рисунке 16 показаны строки кода класса Person, в которых задается условие для вводимого возраста (значения меньше 0 и больше 110 выводится игнорируются), поэтому, чтобы добиться вывода другого значения возраста нужно задать значение в том диапозоне который дан (больше 0 и меньше 110).

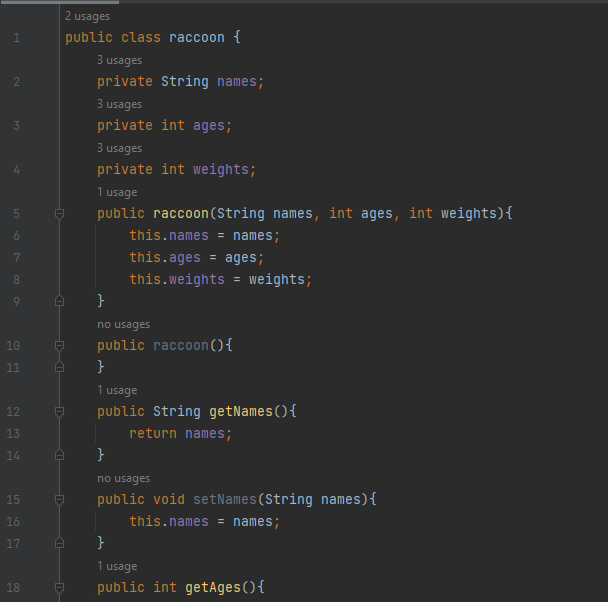


*Рис.17 (вывод другого значения)*

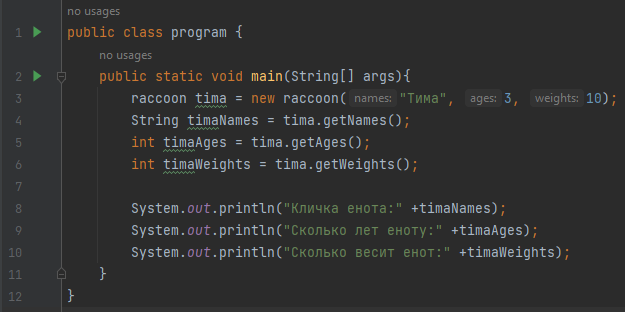
## **Пункт 4**

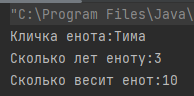
* 1. **Разобраться в коде, приведённом в https://3ty.ru/languages/java/gettery-i-settery-v-java и реализовать его в соответствующем проекте**











*Рис.18-22(реализация кода из данной в задании ссылки)*

## **Пункт 5**

**1.5 Сформулировать преимущества и недостатки инкапсуляции**

**Преимущества:**

***Сокрытие данных:*** у пользователя не будет представления о внутренней реализации класса, не будет видно, как класс хранит значения в переменных.

***Возможность повторного использования:*** инкапсуляция также улучшает повторное использование и ее легко изменить в соответствии с новыми требованиями.

***Включает абстракцию данных, позволяя рассматривать объекты как единое целое.***

***Позволяет легко добавлять новые методы и поля, не затрагивая существующий код.***

**Недостатки:**

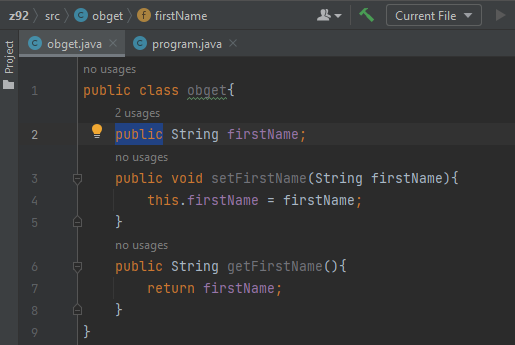
***Может привести к увеличению сложности, особенно при неправильном использовании.***

***Может затруднить понимание того, как работает система.***

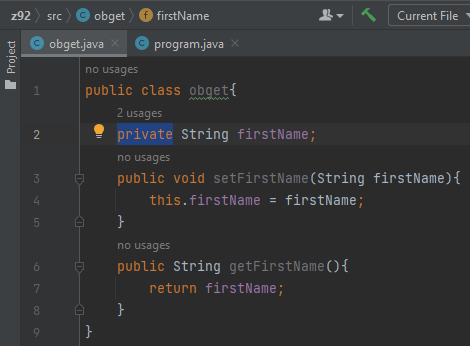
# **Задание 9.2**

## **Пункт 1**

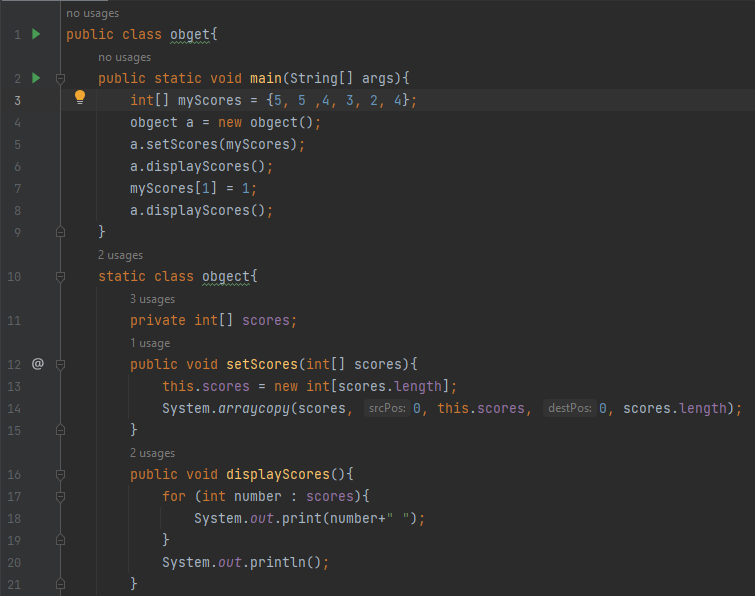
**1. Программно реализовать ошибки и их исправление из раздела 9.3.3**



*Рис.23(ошибка 1)*



*Рис.24(исправление ошибки 1)*





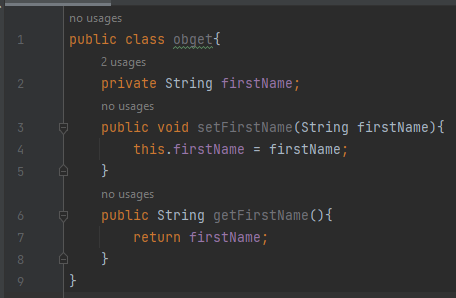
*Рис.25-26*

## **Пункт 2**

**2. Разобрать разделы 9.3.5.1 и 9.3.5.2**

**Геттеры и сеттеры для String**

String — это immutable-тип. Это означает, что после создания объекта этого типа, его значение нельзя изменить. Любые изменения будут приводить к созданию нового объекта String. Таким образом, можно безопасно реализовать геттер и сеттер для переменной String:



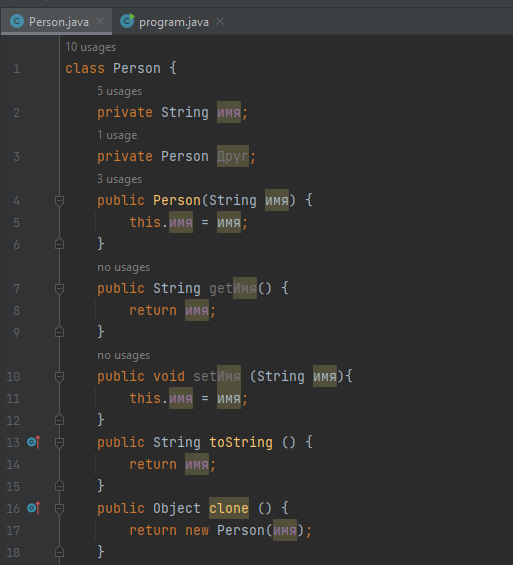
*Рис.27 (пример к использованию геттеров и сеттеров для String)*

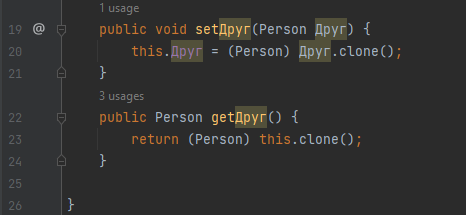
**Геттеры и сеттеры для объектов типа Date**

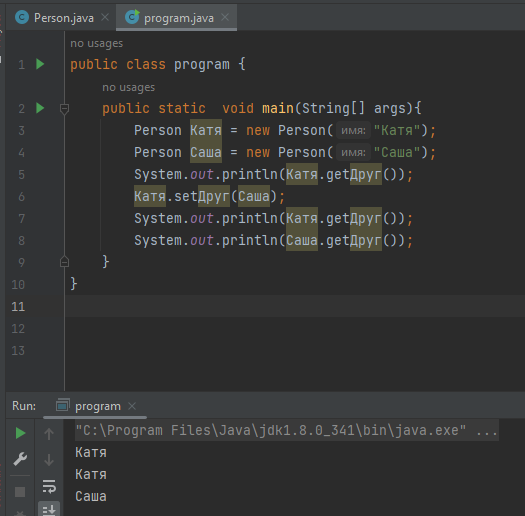
Т.к. объекты класса java.util.Date являются изменяемыми, то внешние классы не должны иметь доступ к их оригиналам. Данный класс реализует метод clone() из класса Object, который возвращает копию объекта, но использовать его для этих целей не стоит.

## **Пункт 3**

**3. Программно реализовать геттеры и сеттеры собственного класса с использованием метода clone().**







*Рис.28-30 (реализация геттеров и сеттеров собственного класса с использованием метода clone())*