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 к лекции №9

Язык Java

Выполнение первой программы на Java

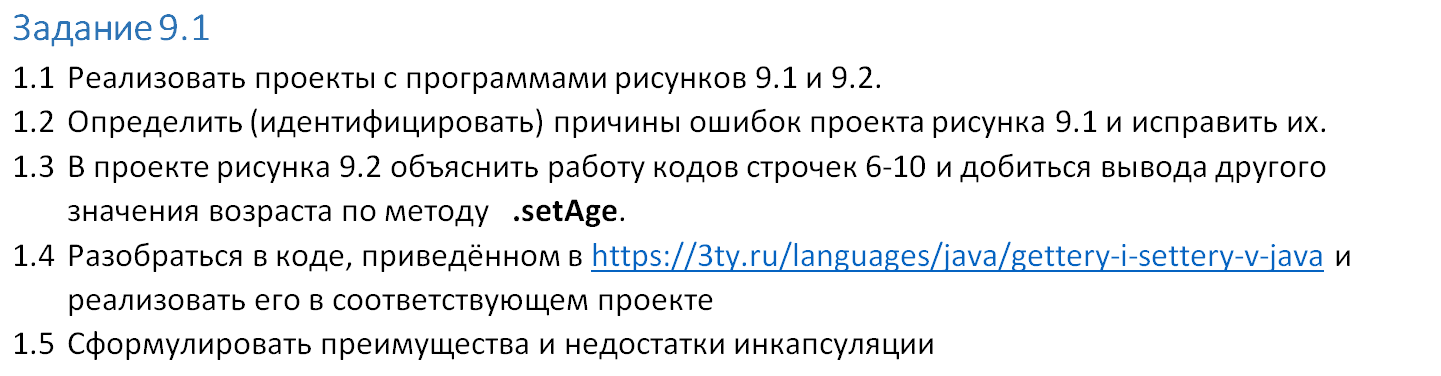
Выполнил: студент группы:

ИБ-321

Епанешников Никита

Нижний Новгород

2023 г.

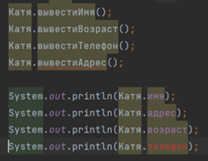


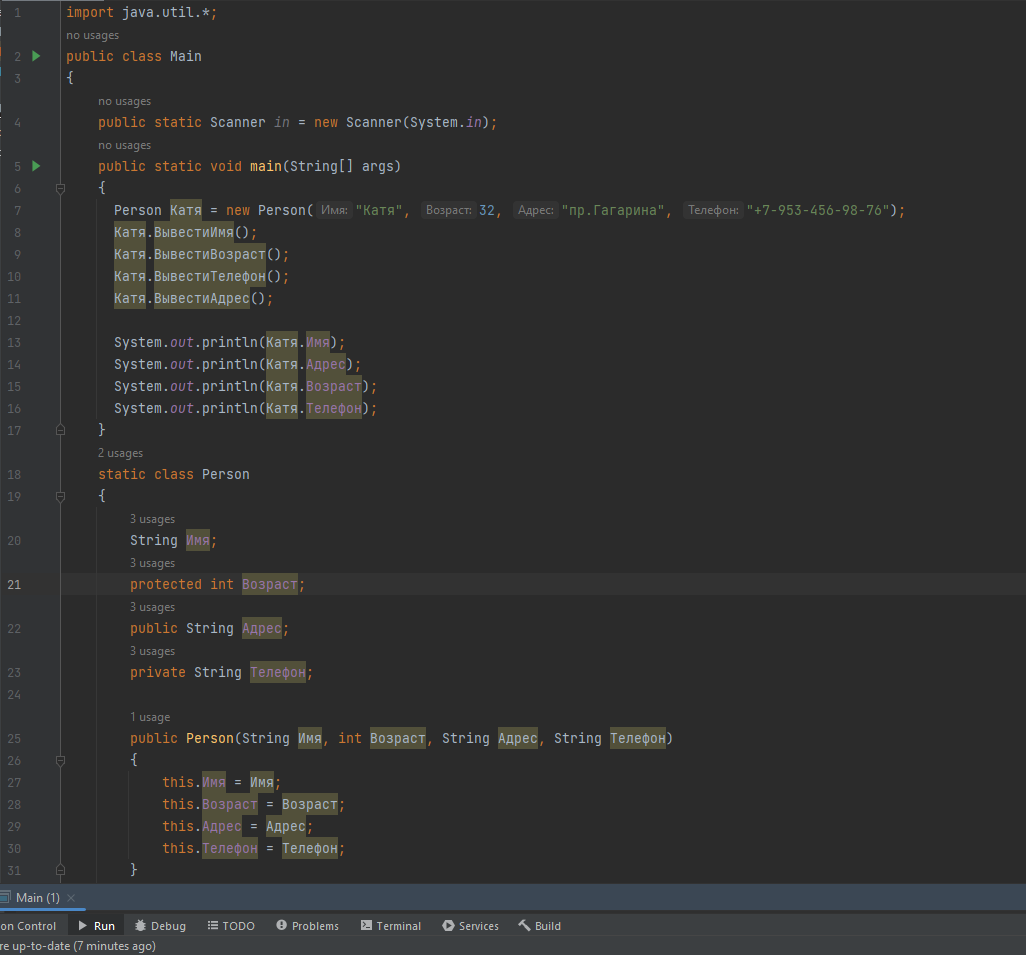
**Задание 1.1**

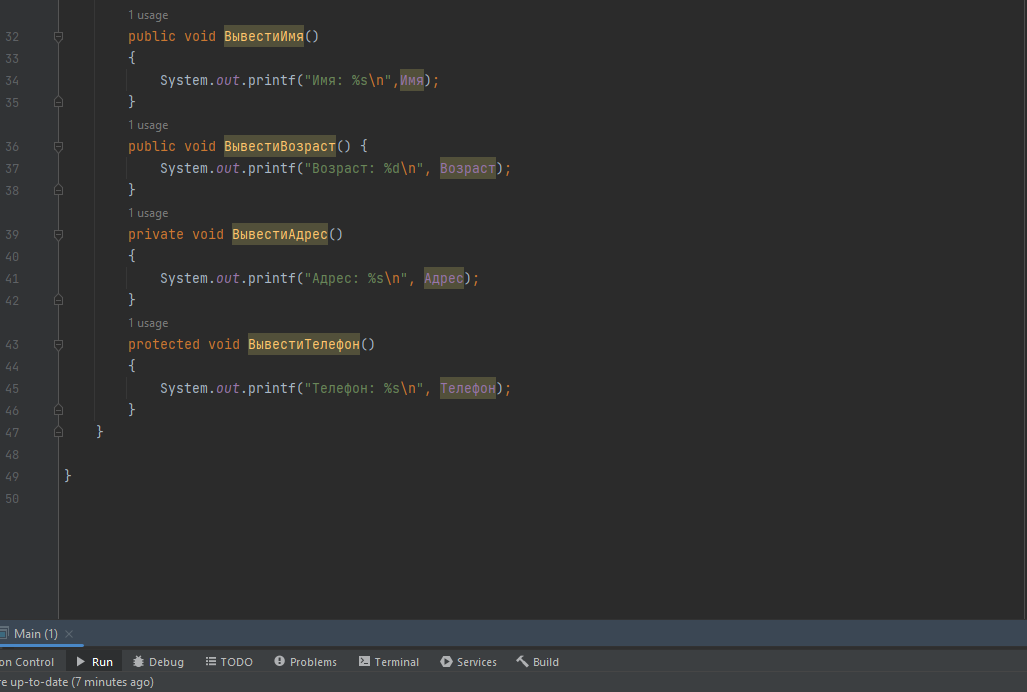
**В рисунке 9.1 были ошибки в следующих строчках:**

 - здесь должен быть статический класс, то есть

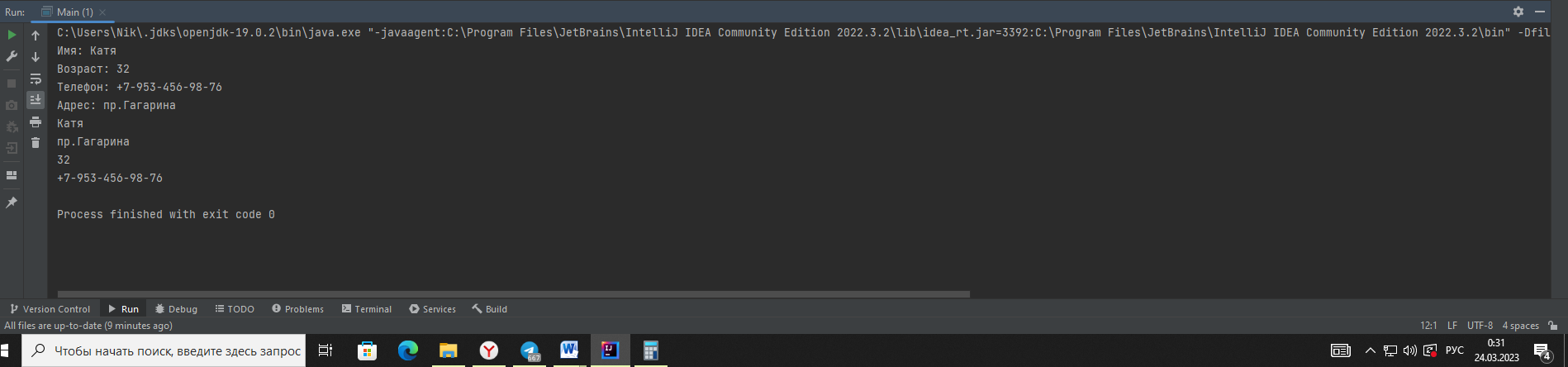
static class Person{

Так же, после точки  нужно писать слова с большой буквы.





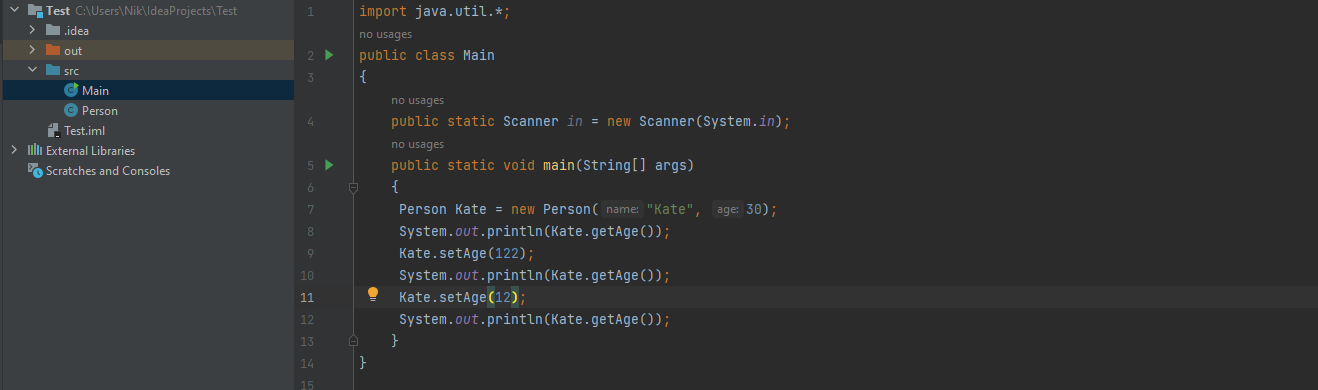
**Вывод в консоль:**

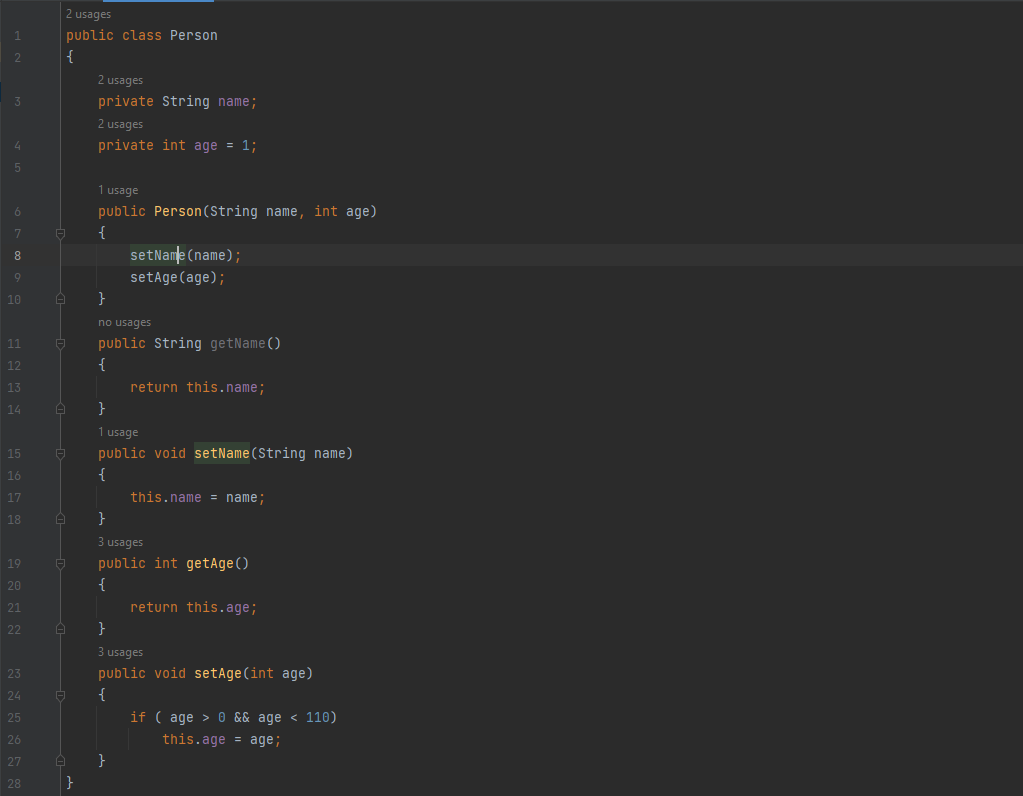


**Задание по рисунку 9.2:**

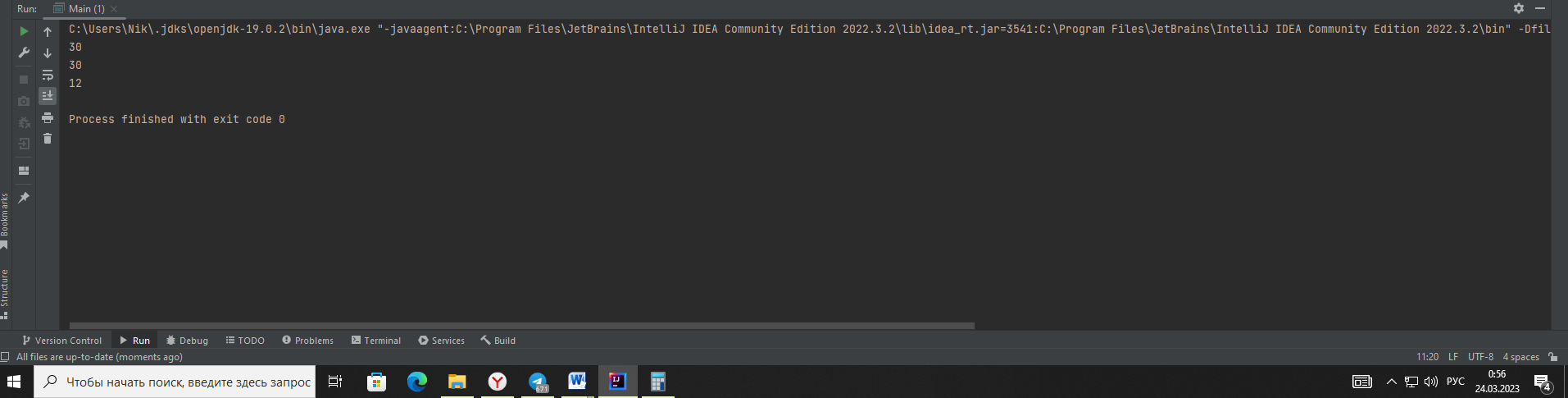
Я создал два файла Person и Main. В Main создаётся класс Person и задаётся возраст, имя.

В Public инициализируются переменные, так же берёт возраст и имя из Main и выводит эти переменные с условием, если age > 0 и < 110, то переменная выведится которую ей задали, но если наоборот, то она выведет то что было последнее, иначе если не удовлетворяло условию в начале, то выводится 1 по умолчанию.





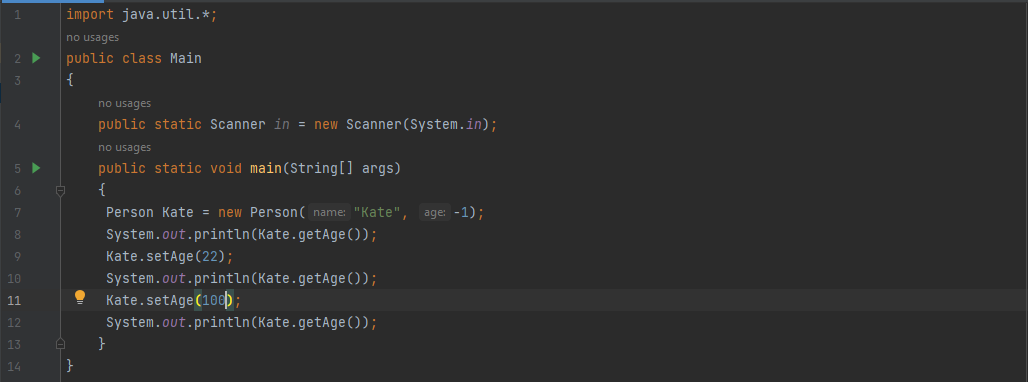
**Вывод в консоль:**



**Задание 1.2**

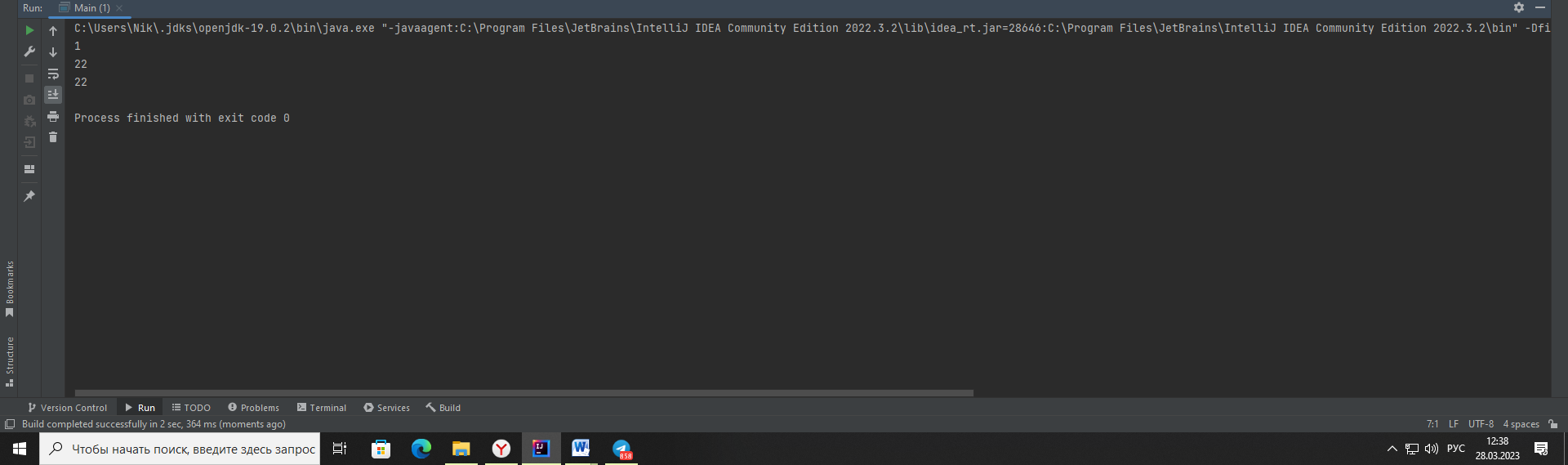
Я объяснил причину ошибки в начале реализации [задания 1.1](#_top).

**Задание 1.3**

****

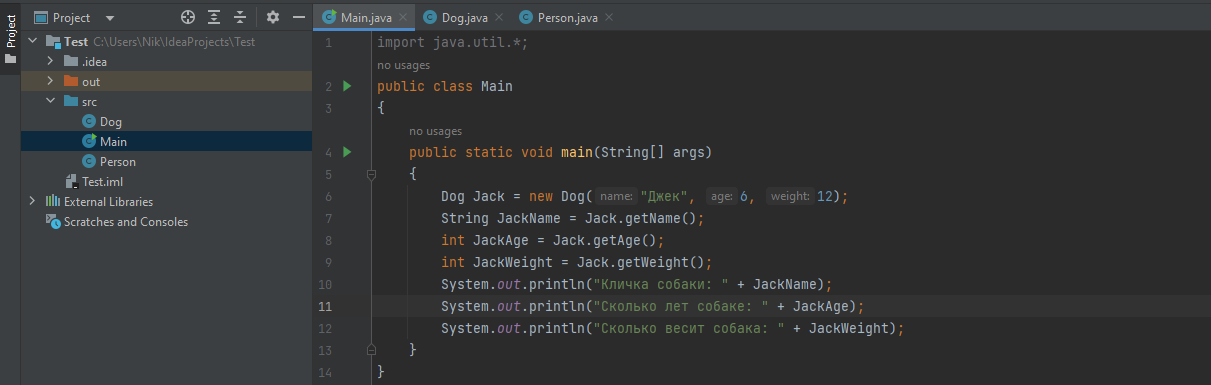
****

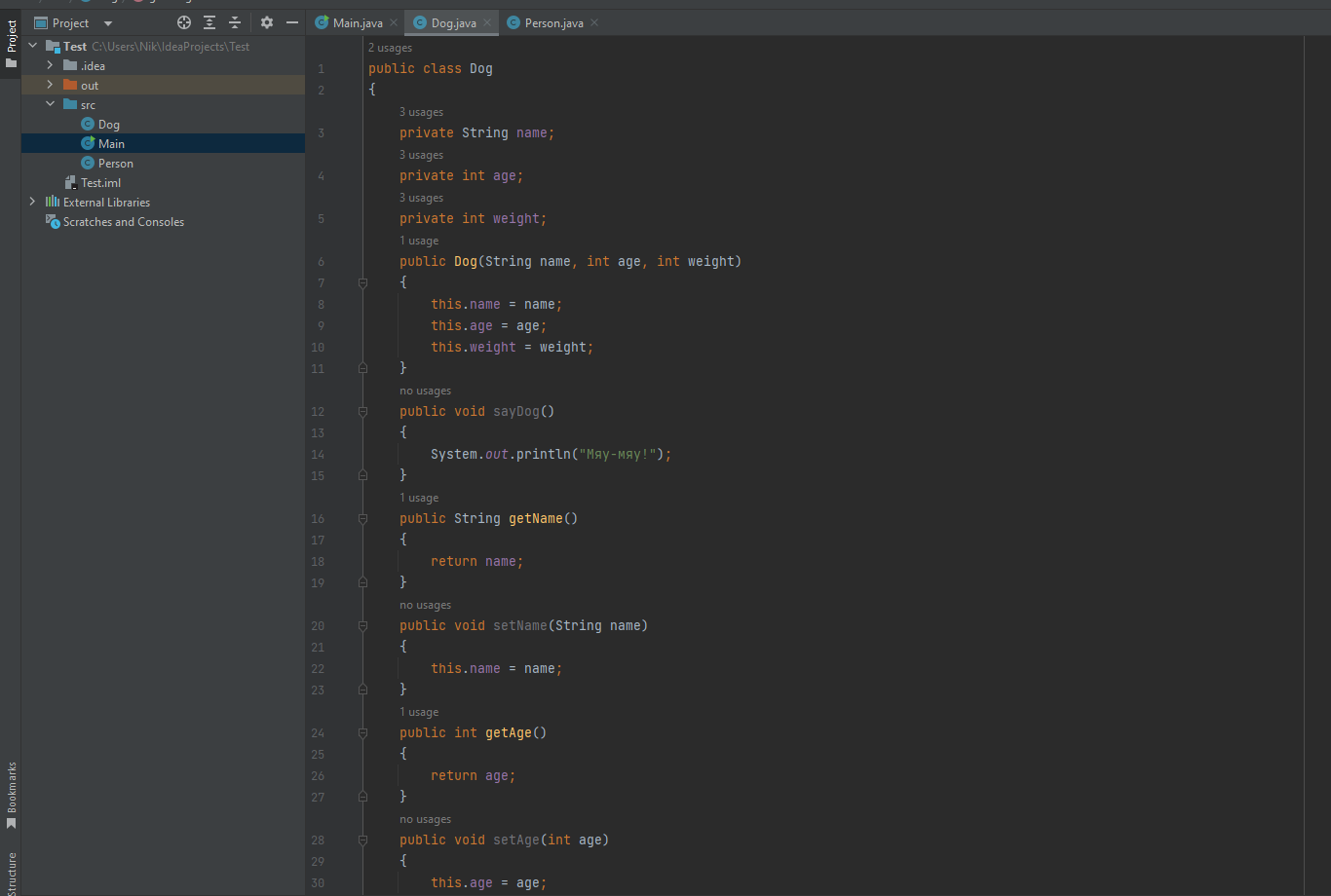
В данных строках мы выводим значения атрибута age при помощи метода getAge(). Далее мы меняем значение атрибута при помощи метода setAge() и снова выводим значение атрибута. В файле main я указал возраст -1, но так как у нас стоит условие, что возраст не может быть меньше 0 и больше 110, то он принимает значение, которое стоит по умолчанию и выводит 1. Далее изменяю возраст на 22 и выводится это значение, если в третьем Kate.setage указать значение 150, компилятор выведет то значение, которое было до него, а это - 22. Все свои слова документировал в примера выше. Ответы ниже.

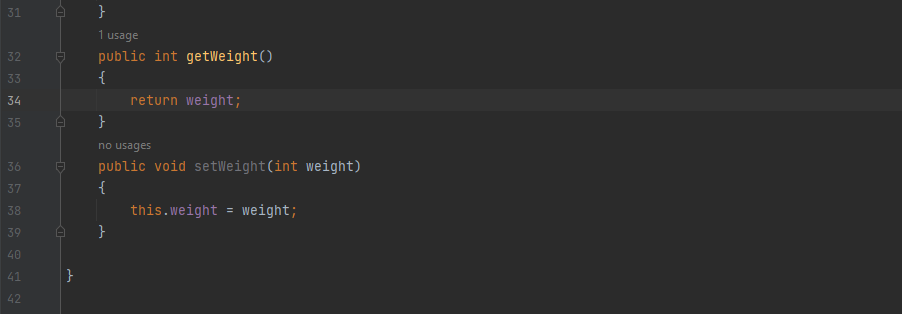
****

**Задание 1.4**

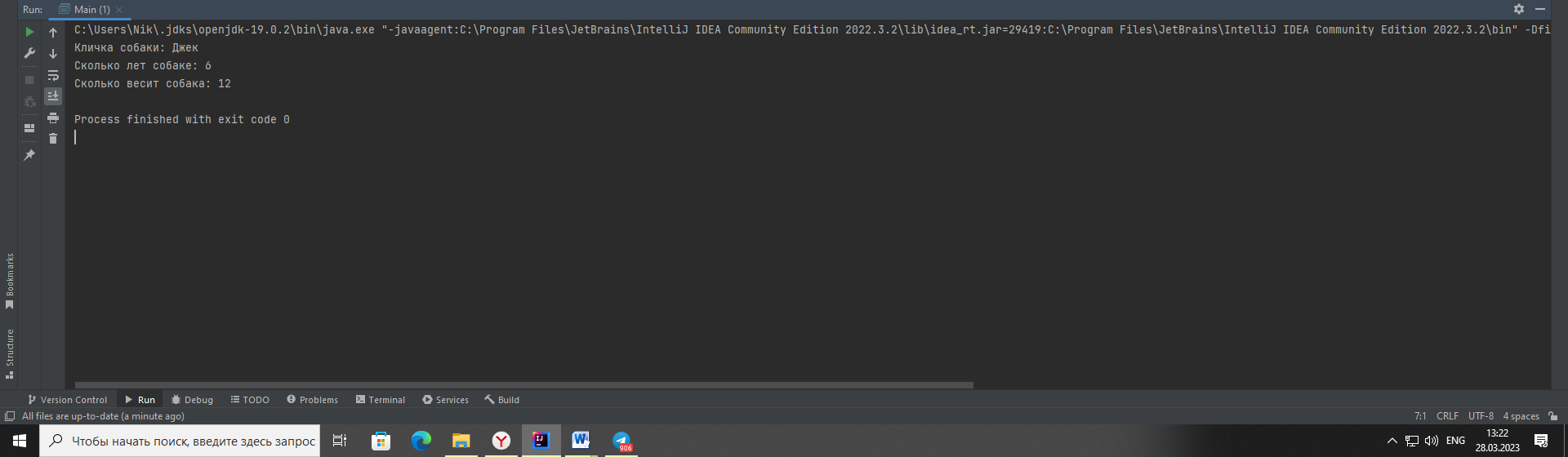
В public class main мы заходим в первую же строчку и попадаем в public class dog. Далее мы идем по коду заполняя элементы name, age, weight с помощью this. Потов мы выходим из этой функции и попадаем обратно в class main, заходим в 7 строчку и принимаем значение getName() Джек. Так же происходит с getAge(), заполняя значение age = 6. Потом Weight со значением со значением 12. Напомню, все значения примаются из class dog! Ну и в конце мы просто выводим эти значения с помощью System.out.println()





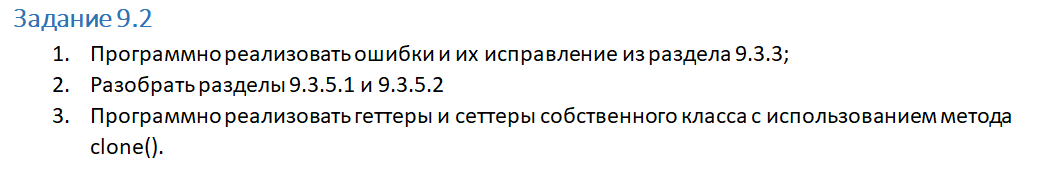


Вывод в консоль:



**Задание 1.5**

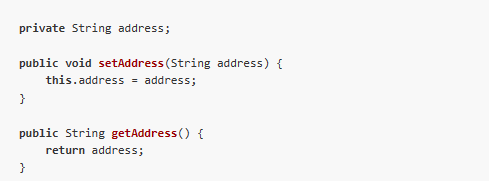
Главным преимуществом инкапсуляции является защищенность данных, а недостатком понижение скорости работы



## **Задание 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, как и для примитивных типов, можно безопасно реализовать геттер и сеттер для переменной 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.

Поскольку Date не является окончательным классом, нет га­рантии, что метод clone() возвратит объект, класс которого именно java.util.Date: он может вернуть экземпляр ненадежного подкласса, созданного специально для нанесения ущерба. Такой подкласс может, например, записы­вать ссылку на каждый экземпляр в момент создания последнего в закрытый статический список, а затем предоставить злоумышленнику доступ к этому списку. Чтобы это исправить следует каждый раз создавать новый экземпляр класса:

