|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования РФ  Федеральное государственное автономное образовательное учреждение высшего образования  «Пермский государственный национальный исследовательский университет» | | |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по лабораторной работе №3  по дисциплине «Языки программирования»  Вариант 4 | | |
|  | | Работу выполнил  студент группы ПМИ-7,8-2023 2 курса  Семенов И.Н.  «\_\_\_» \_\_\_\_\_\_\_\_2024 г. |
| Работу проверил  Ракина В.Д  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г. |
| Пермь 2024 | | |

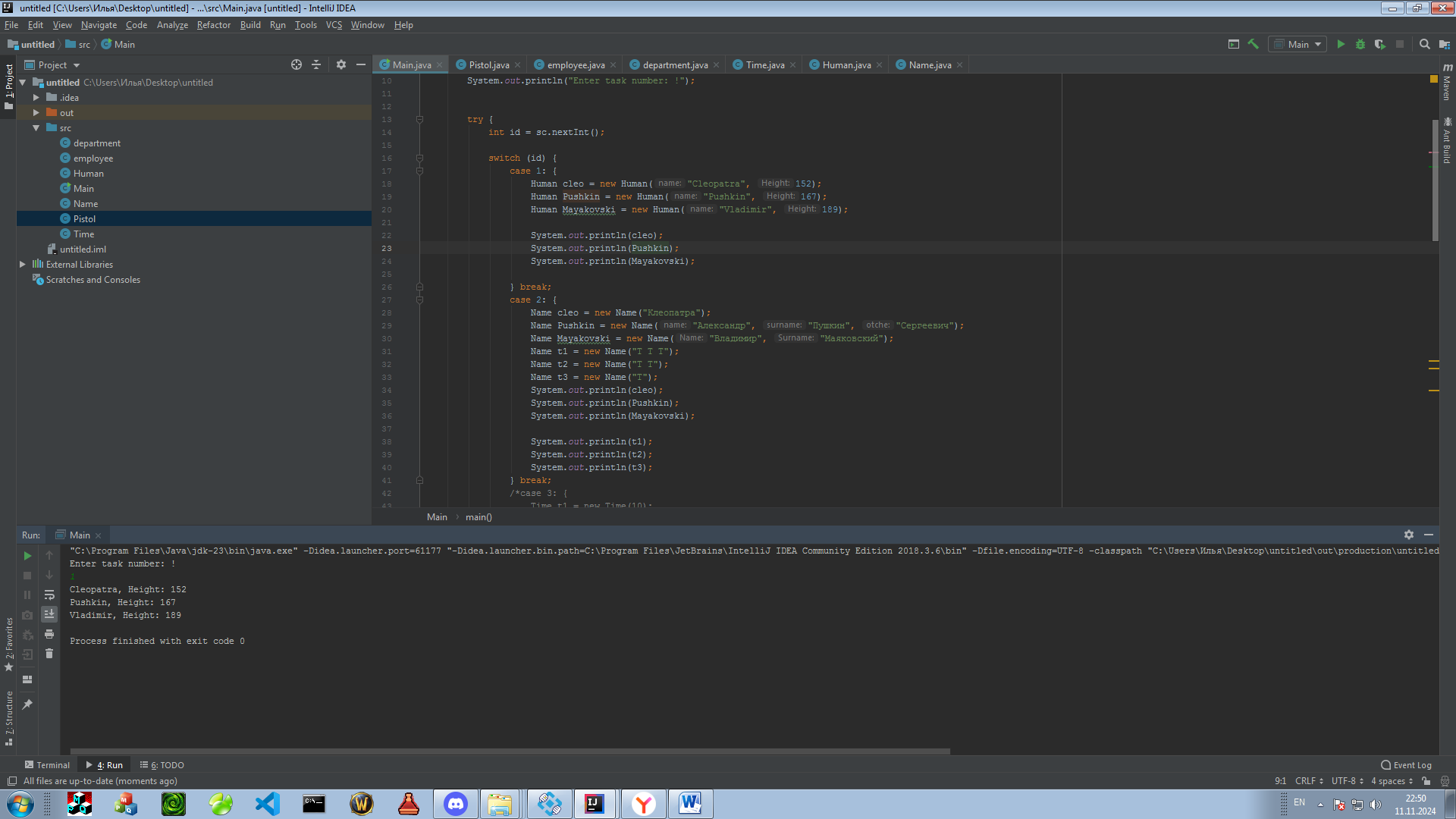
СОДЕРЖАНИЕ

[Задание 1 4](#_Toc182259343)

[Текст задания 4](#_Toc182259344)

[Алгоритм решения 4](#_Toc182259345)

[Тестирование 5](#_Toc182259346)

[ 5](#_Toc182259347)

[Код программы 5](#_Toc182259348)

[Задание 2 5](#_Toc182259349)

[Текст задания 5](#_Toc182259350)

[Алгоритм решения 6](#_Toc182259351)

[Тестирование 6](#_Toc182259352)

[Код программы 6](#_Toc182259353)

[Задание 3 7](#_Toc182259354)

[Текст задания 7](#_Toc182259355)

[Алгоритм решения 7](#_Toc182259356)

[Тестирование 8](#_Toc182259357)

[Код программы 8](#_Toc182259358)

[Задание 4 9](#_Toc182259359)

[Текст задания 9](#_Toc182259360)

[Алгоритм решения 9](#_Toc182259361)

[Тестирование 10](#_Toc182259362)

[Код программы 10](#_Toc182259363)

[Задание 5 11](#_Toc182259364)

[Текст задания 11](#_Toc182259365)

[Алгоритм решения 11](#_Toc182259366)

[Тестирование 11](#_Toc182259367)

[Код программы 11](#_Toc182259368)

[Задание 6 12](#_Toc182259369)

[Текст задания 12](#_Toc182259370)

[Алгоритм решения 12](#_Toc182259371)

[Тестирование 13](#_Toc182259372)

[Код программы 13](#_Toc182259373)

# Задание 1

## Текст задания

Измените сущность Пистолет из задачи 1.5.1. Модификация предполагает внесение следующих

дополнительных требований:

 Имеет максимальное количество патронов. Максимальное количество устанавливается во

время создания пистолета и не может быть изменено позднее. У пистолета можно узнать,

какое максимальное количество он вмещает.

 Может быть перезаряжен. Для перезарядки необходимо передать пистолету число,

которое будет означать количество заряжаемых патронов. Если передано отрицательное

число, необходимо выбросить ошибку, объясняющую, что отрицательного числа

патронов быть не может. Если передано слишком большое число патронов – необходимо

лишние вернуть.

 Может быть разряжен. Это приводит к обнулению патронов в пистолете и возврате

нужного числа пользователю.

 Можно узнать сколько сейчас заряжено патронов.

 Можно узнать заряжен он или разряжен.

Создайте пистолет вместимостью 7, зарядите три патрона, выстрелите из него пять раз, затем

зарядите в него 8 патронов, выстрелите еще 2 раза, разрядите его, сделайте контрольный

выстрел.

Если все выполнено верно, то должно быть выведено: Бах! Бах! Бах! Клац! Клац! Бах! Бах! Клац!

## Алгоритм решения

Изменить сущность, добавить поле MaxRounds, изменить конструктор, добавить геттер для поля. Имплементировать функцию перезарядки, использовать throw для выкидывания ошибки если кол-во заряжаемых патронов меньше 0. Создать геттер для поля с кол-вом патронов.

## Тестирование

## 

## Код программы

https://github.com/SemenovIlya12/Jab/tree/Lab2

# Задание 2

## Текст задания

Непустые Имена.

Измените сущность Имя из задачи 1.4.5. Гарантируйте, что:

 Как минимум один параметр будет иметь не null значение и не пустую строку.

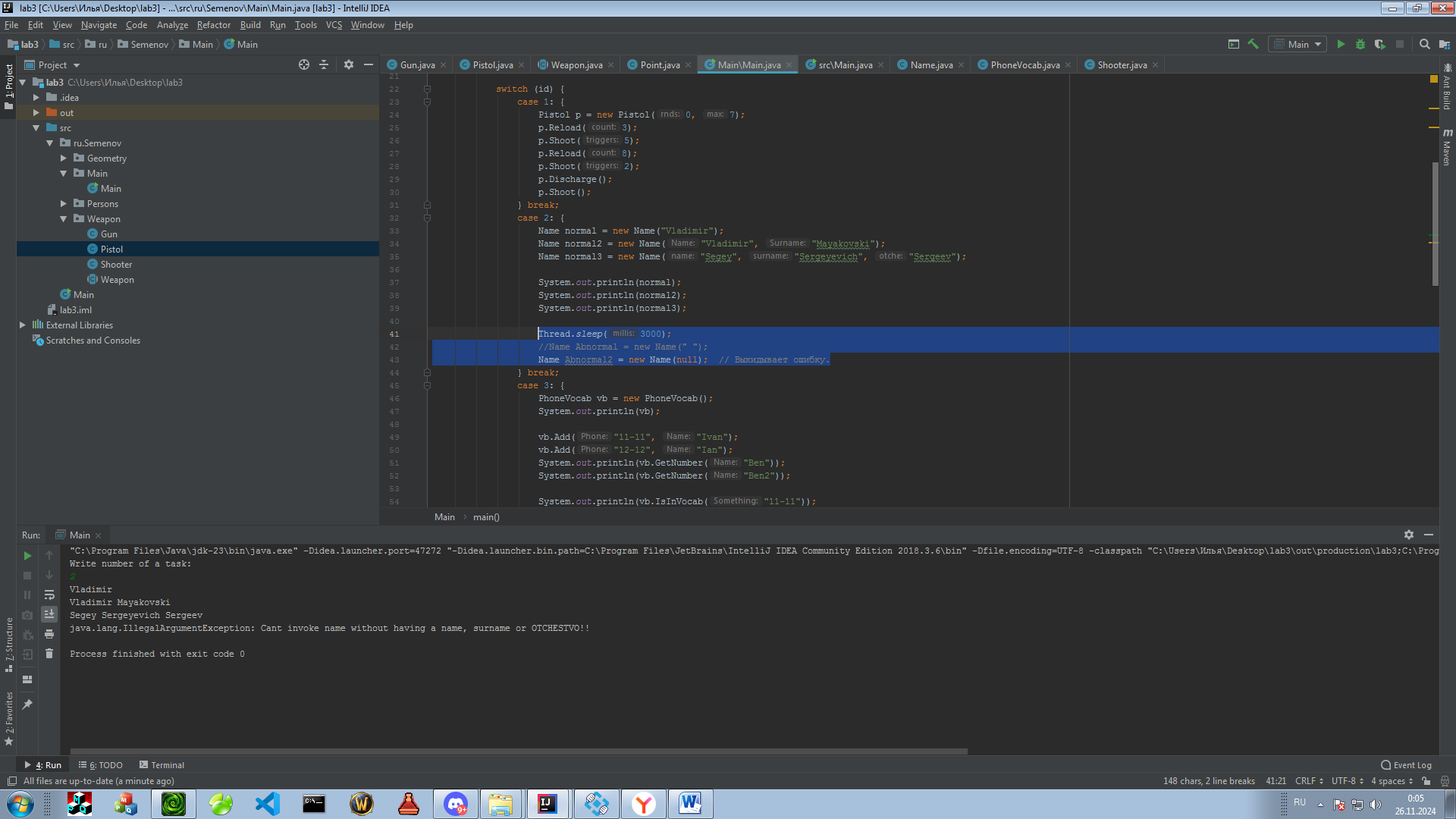
 Имя неизменяемо.

Продемонстрируйте работоспособность решения на примерах

## Алгоритм решения

Сделать проверку в конструкторе на наличие хотя бы 1 параметра, отличного от null/пустой строки. изменить поле имя на private final.

## Тестирование



## Код программы

Ссылка на githab

# Задание 3

## Текст задания

Необходимо разработать сущность ТелефонныйСправочник, который будет использоваться для

создания справочника со списком телефонов и имен, с целью поиска телефона по имени. Будем

считать достаточным хранение пар телефон-строка, например “89003337788 - Вася”.

Состояние сущности описывается набором пар “телефон – имя”, причем и телефон и имя являются

строками. В целях упрощения задачи будем считать, что одному имени всегда соответствует один

телефон, и один телефон соответствует одному имени.

Инициализация сущности выполняется без аргументов.

Поведение сущности описывают следующие действия:

 Добавление новой пары “телефон – имя” (как два отдельных аргумента). Конкретное

местоположение добавленной пары не важно. Если добавляемое имя уже есть в списке, то

имеющийся телефон заменяется на новый, и затем старый телефон возвращается из

метода.

 Удаление значения. Для удаления необходимо указать имя удаляемого контакта, после

чего удаляется вся пара.

 Получение значений. Для получения необходимо указать имя контакта, тогда

возвращается его телефон.

 Может быть приведен к строке. Строка должна представлять собой полный перечень всех

хранимых пар.

 Можно проверить есть ли конкретный телефон или конкретное имя в списке.

 Можно узнать текущее количество контактов.

 Можно запросить в виде массива:

o Все пары

o Все телефоны

o Все имена

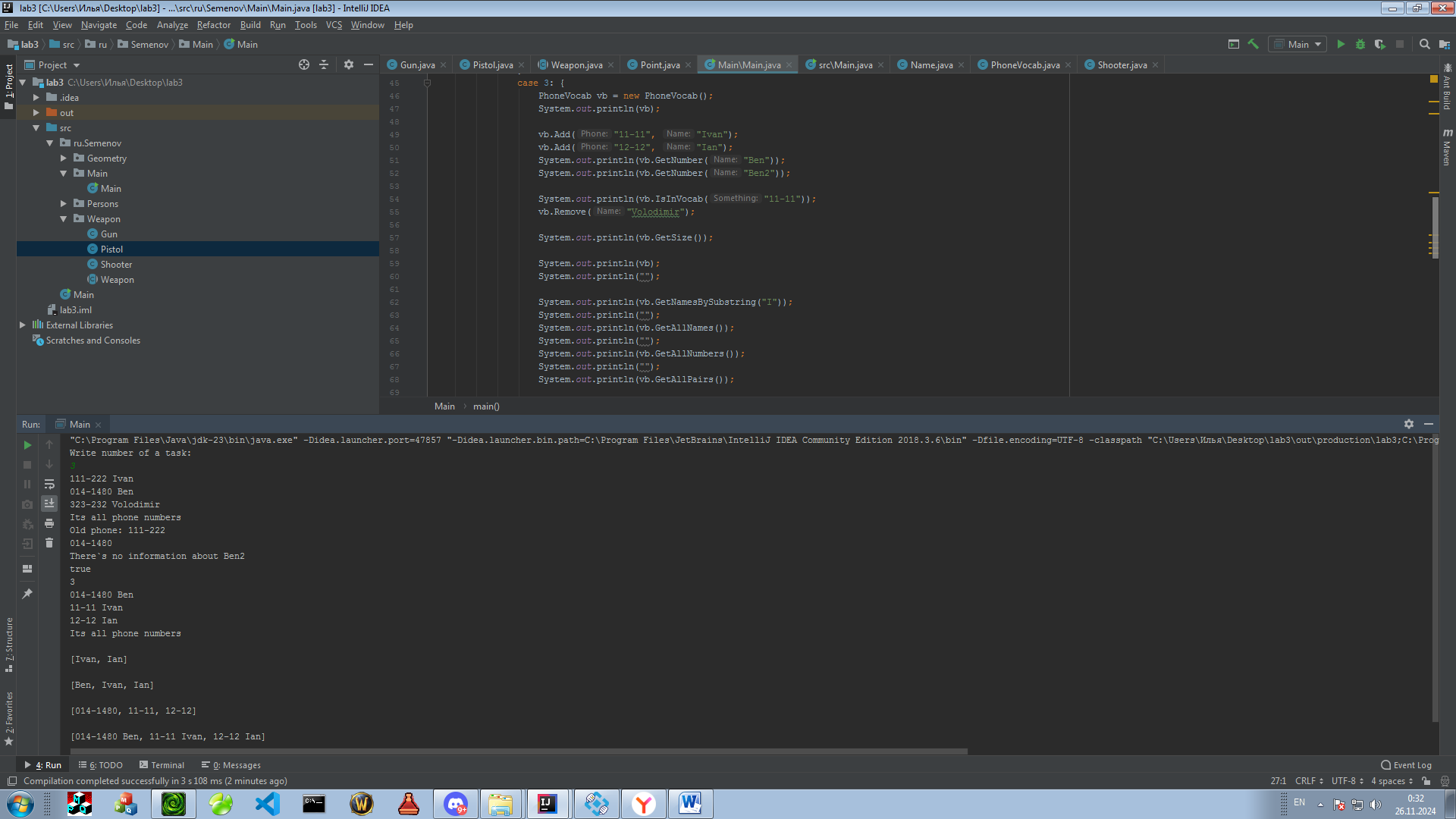
 Можно получить массив всех имен, указав часть названия (начиная с первого символа).

Продемонстрируйте работоспособность решения на примерах.

## Алгоритм решения

Создаем сущность с полем PhoneVocab – тип данных Map, которая будет хранить пары ключ-значение, <String, String>. В конструкторе заполняем словарь. Создаем 2 приватных метода, которые мы будем использовать только внутри самого класса – GetNumberByName, и флаг IsNameExists. Создаем метод для добавления номера, в случае когда номер уже существует в словаре, заменяем и возвращаем старый телефон. Создаем метод для удаления из словаря. Метод получения размерности словаря, метод для проверки существования имени/номера в словаре используя contains(). Метод для получения имени по телефону. Методы получения массивов – имен с подстрокой, имен, телефонов, всех пар.

## Тестирование



## Код программы

Ссылка на githab

# Задание 4

## Текст задания

Создайте такой подвид сущности Пистолет из задачи 2.1.5, которая будет совпадать с

ней во всех отношениях, кроме следующего:

 Имеет скорострельность (целое число, неизменяемое) которое обозначает количество

выстрелов в секунду, поддерживаемое данным автоматом. Скорострельность всегда

положительное число.

 При вызове Стрелять количество выстрелов соответствует скорострельности (например,

при скорострельности 3 выводится три строки с текстом выстрела).

 Умеет Стрелять N секунд, что приводит к количеству выстрелов равному N умноженное

на скорострельность.

 Инициализация может быть выполнены следующими способами:

a) Без параметров. Скорострельность 30, вместимость 30.

b) С указанием максимального числа патронов. Скорострельность будет равна

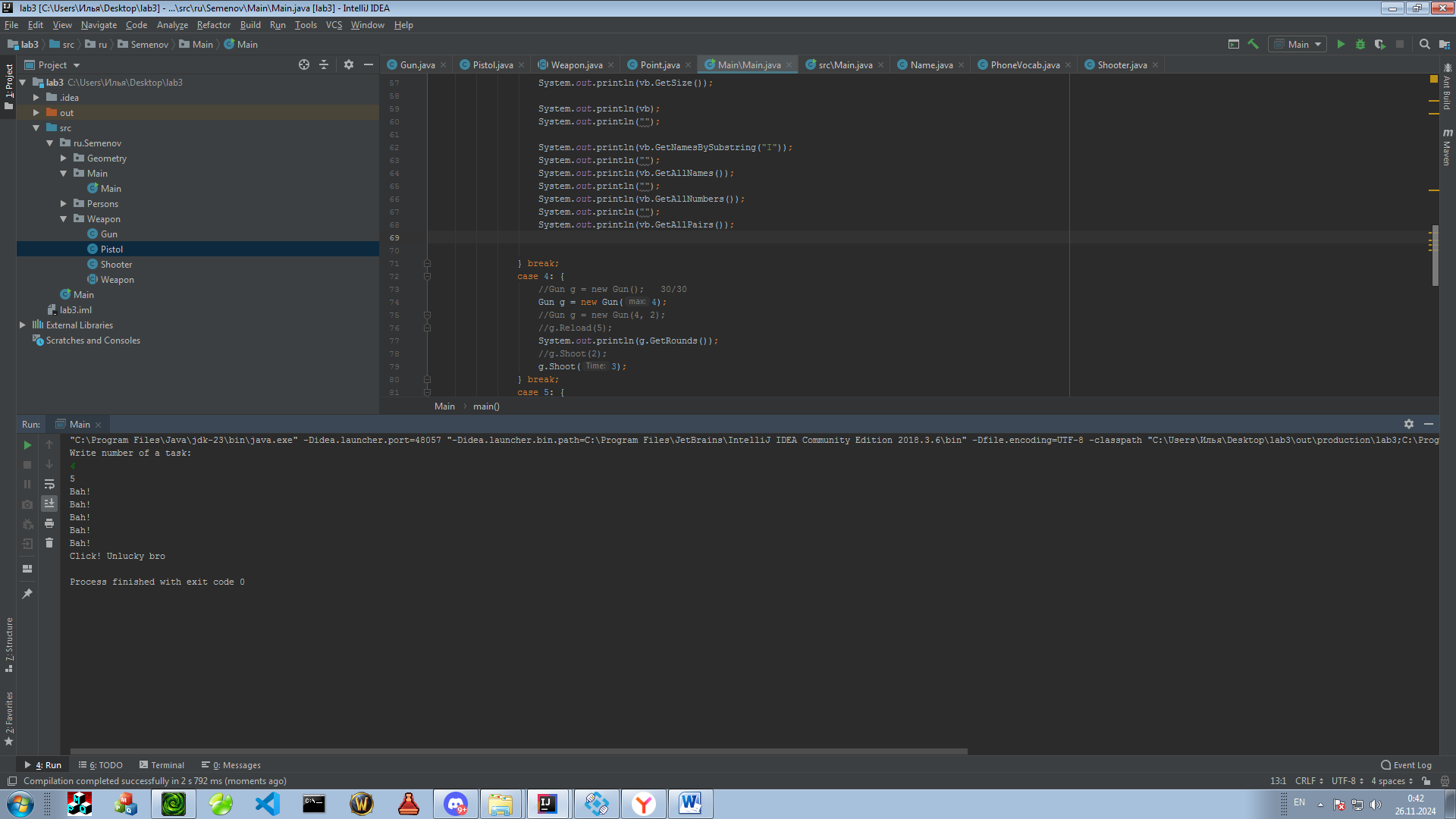
половине обоймы

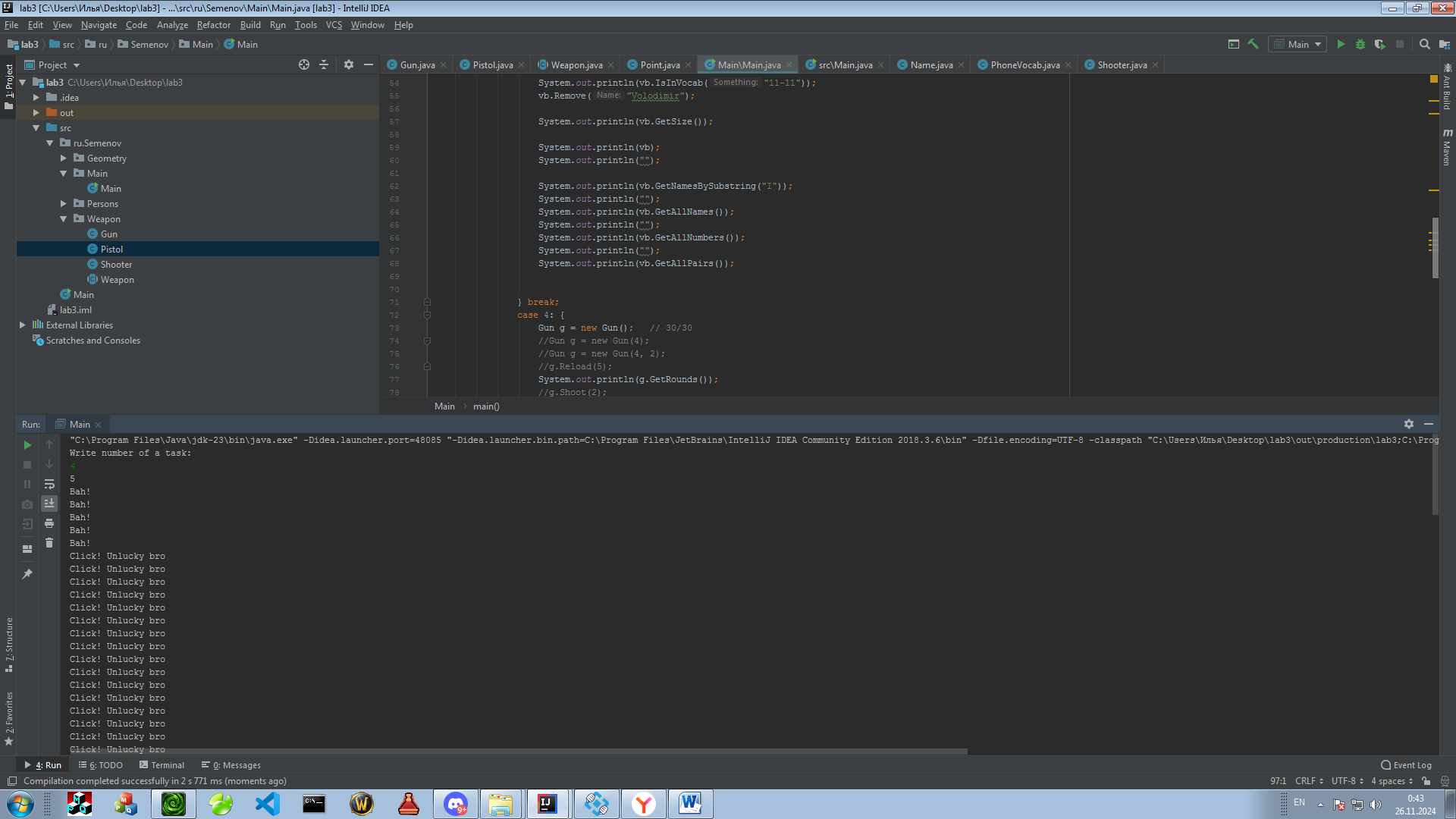
c) С указанием максимального количества патронов в обойме и скорострельности.

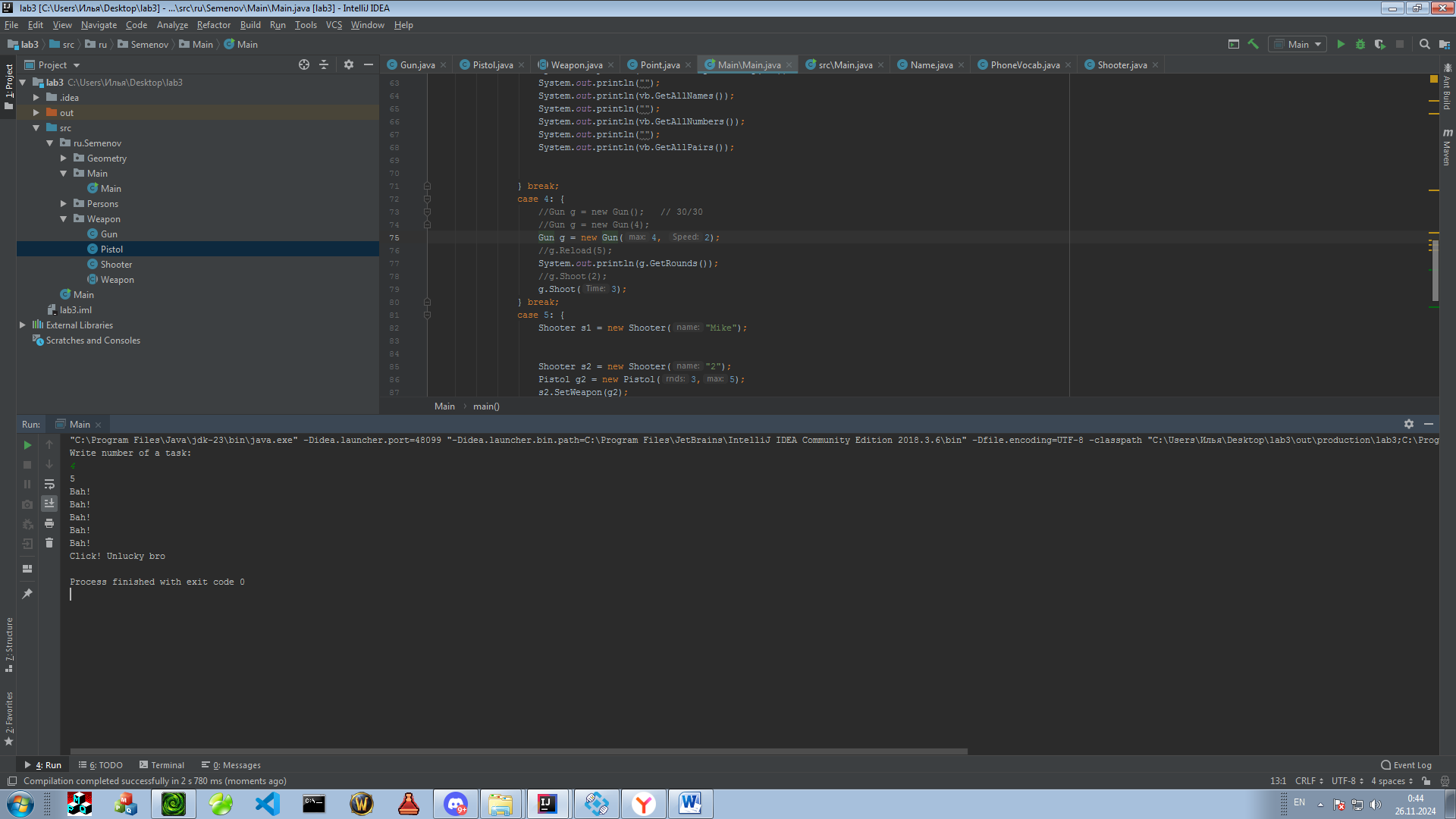
## Алгоритм решения

Создаем сущность автомат, которая будет наследоваться от пистолета. Создаем поле private final int speed – скорострельность. Перегружаем конструктор, т.к у нас есть несколько вариантов инициализации. Перезаписываем метод Shoot с учетом скорострельности, также перегружаем его вариацией, когда нам дано время в секундах, в течении которого автомат стреляет.

## Тестирование







## Код программы

Ссылка на githab

# Задание 5

## Текст задания

Измените сущность Пистолет, полученную в задаче 2.1.5 таким образом, чтобы она наследовалась

от класса Weapon описанного на рисунке:

## Алгоритм решения

Поле с кол-вом патронов больше не требуется, так как уже наследуется от weapon, вызываем конструктор суперкласса в конструкторах пистолета.

## Тестирование

-

## Код программы

Ссылка на githab

# Задание 6

## Текст задания

Создайте сущность Стрелок, которая описывается:

 Имя, строка

 Оружие, из задачи 2.4.1.

 При создании объекта необходимо указать ему имя

 Имя и оружие можно поменять и получить в любой момент без ограничения.

Основная способность Стрелка - умение стрелять. Если оружие есть, то выстрел происходит по

правилам оружия, если его нет – то выводится текст “не могу участвовать в перестрелке”.

Создайте трех стрелков: одного без оружия, одного с пистолетом и одного с автоматом, и пусть

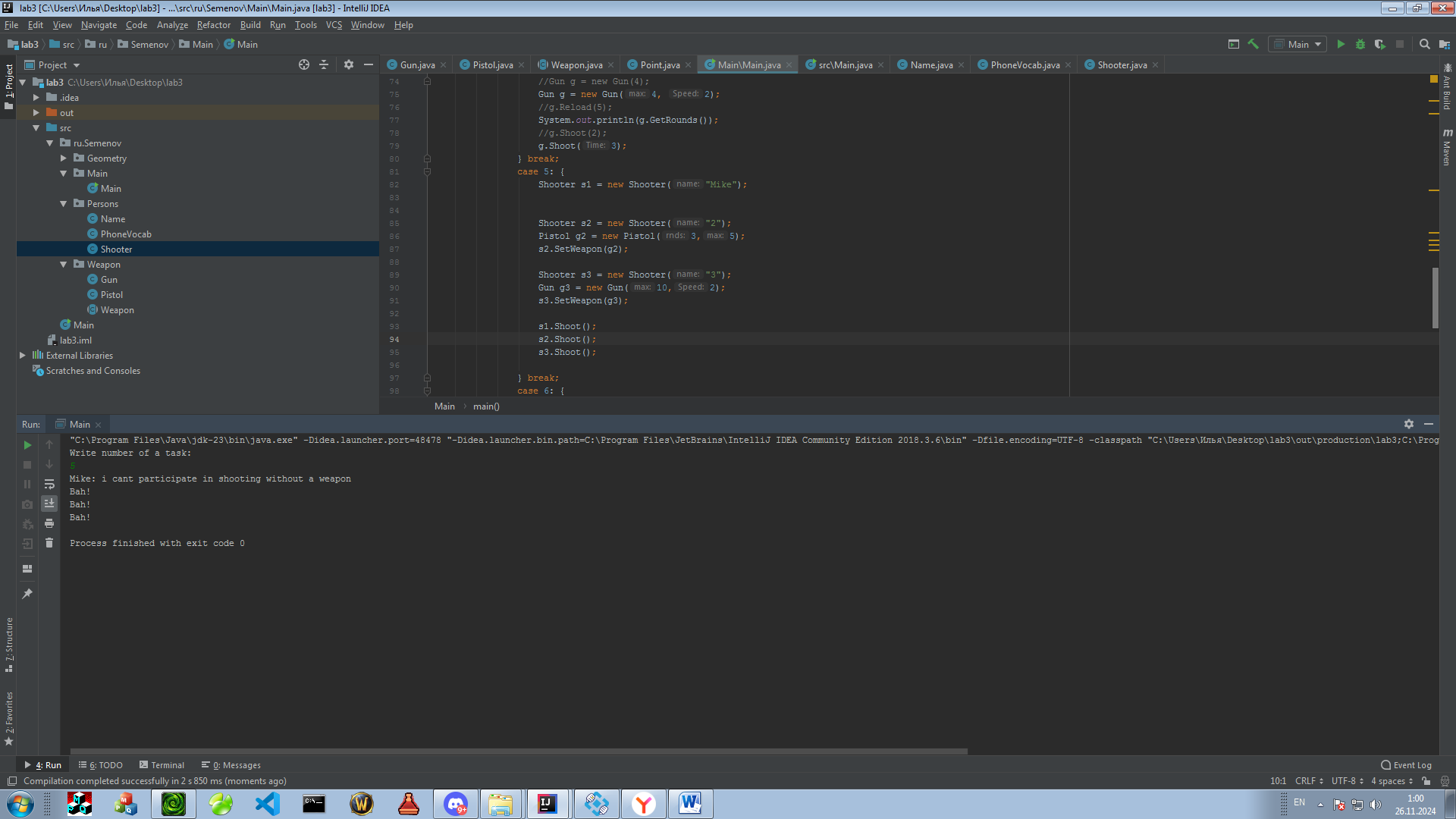
каждый из них выстрелит.

## Алгоритм решения

Создаем сущность стрелок, поля String name, Weapon weapon = null.

Создаем конструктор, в котором даем имя стрелку. Создаем сеттеры для имени и оружи.я. Создаем метод для стрельбы, в котором проверяем, есть ли у стрелка оружие.

## Тестирование



## Код программы

Ссылка на githab

**Задание 7**

**Текст задания**

Измените сущность Точка из задачи 1.4.1. Переопределите метод сравнения объектов по

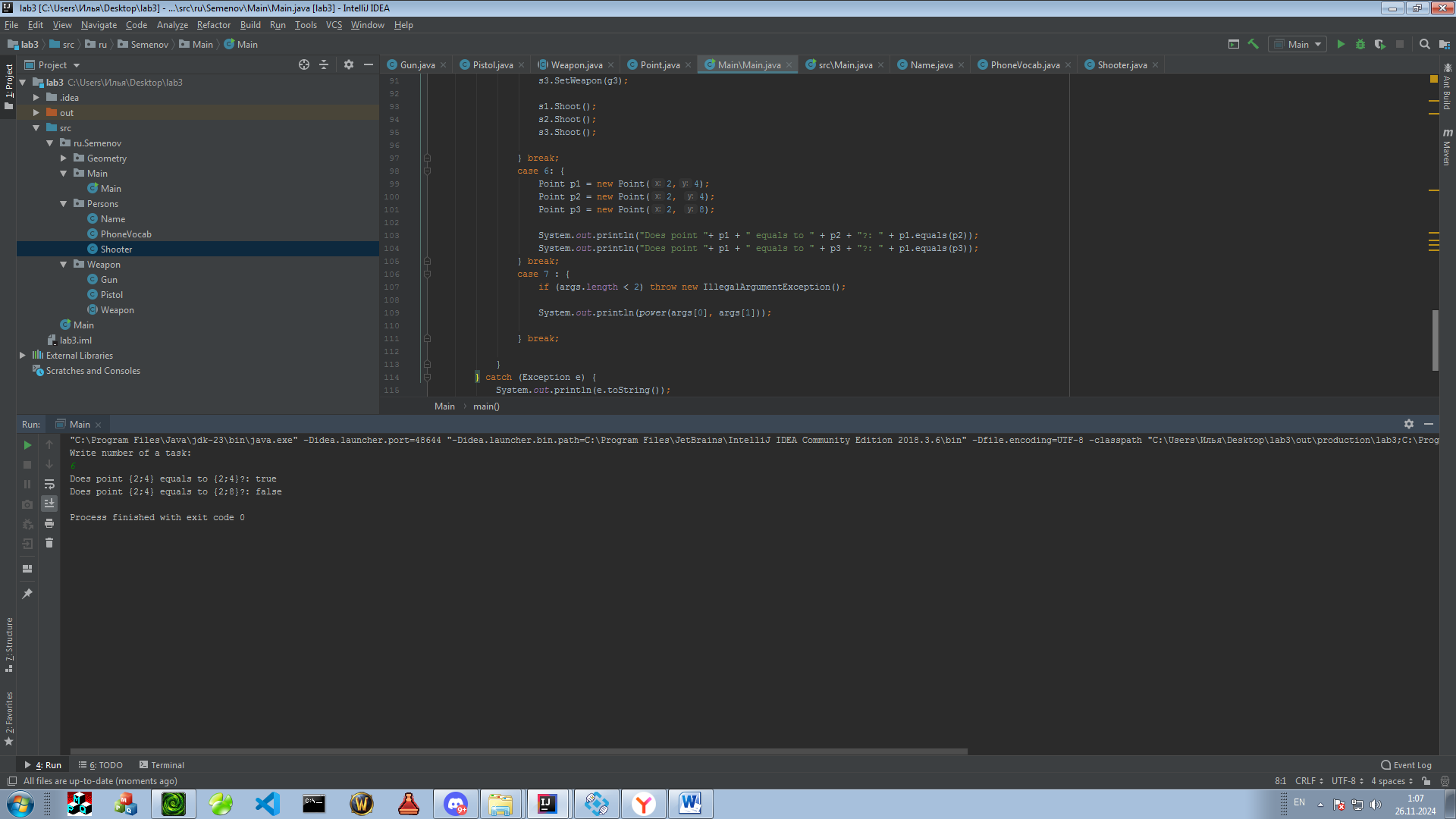
состоянию таким образом, чтобы две точки считались одинаковыми тогда, когда они

расположены в одинаковых координатах.

**Алгоритм решения**

Перезаписываем метод Equals, проверяем на схожесть классов, если не схожи – return false, иначе сравниваем координаты и возвращаем результат.

**Тестирование**



**Код программы**

Ссылка на githab

**Задание 8 - Пакеты**

**Текст задания**

Создайте метод принимающий две строки, в которых будут записаны числа X и Y. Возвращает

метод результат возведения X в степень Y. Для преобразования строки в число следует

использовать метод Integer.parseInt, а для возведения в степень метод Math.pow. Вызовите

разработанный метод передав туда параметры командной строки полученные точкой входа в

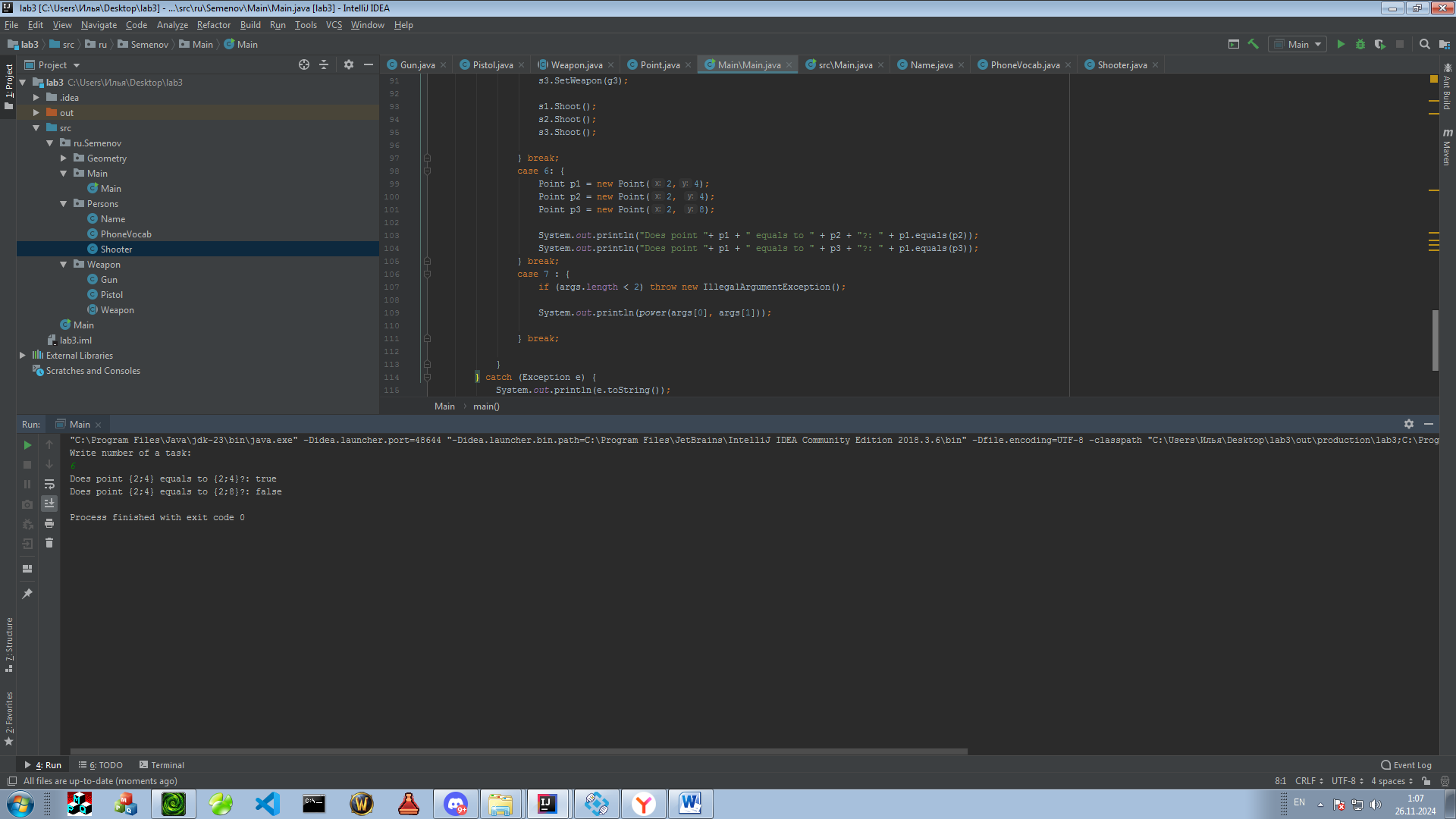
программу. Реализуйте метод так, что бы для возведения в степень и преобразования строки

использовались короткие имена статических методов.

**Алгоритм решения**

Реализуем метод возведения X в степень Y (Math.pow), В точке входа (main) проверяем количество аргументов командной строки, если их 2 и более, то первые две передаем в метод.

**Тестирование**



**Код программы**

Ссылка на githab