|  |  |  |
| --- | --- | --- |
| 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 №4  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 | | |

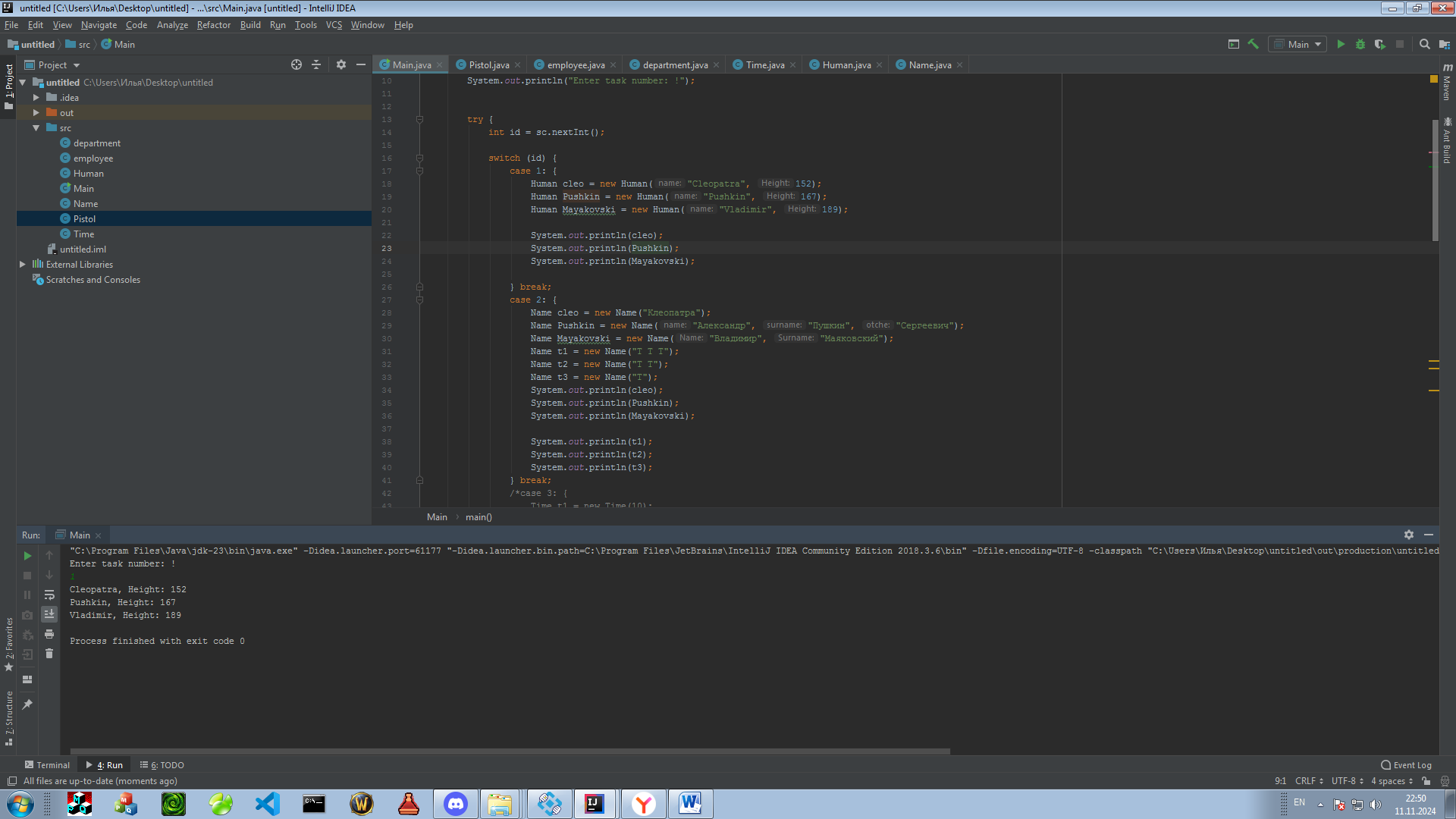
СОДЕРЖАНИЕ

[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

Обобщенная коробка.

Создайте сущность Коробка, которая обладает следующими характеристиками:

 Может хранить один произвольный объект в один момент времени.

 Объект можно получить и разместить на хранение в любой момент времени.

 Если объект забирают из коробки – ссылку на этот объект необходимо обнулить.

 Если объект кладут в коробку, но она не пуста – необходимо выкинуть исключение.

 Имеет метод проверки на заполненность.

 Методы класса должны работать с тем типом данных, который был указан во время

создания объекта

Создайте Коробку которая может хранить целочисленное значение, разместите туда число 3.

Передайте Коробку в какой-либо метод, извлеките значение, и выведите его на экран.

## Алгоритм решения

Создаем обобщенный класс Box<T>. В классе Box<T> объявляем приватное поле item типа T, которое будет хранить объект внутри коробки. Создаем конструктор по умолчанию, который инициализирует поле item значением null.

Создаем метод SetItem(T item), который принимает объект типа T и пытается поместить его в коробку. Внутри метода проверяем, не является ли поле item равным null. Если item не равно null, значит коробка уже заполнена, и мы выбрасываем исключение IllegalStateException с сообщением "Box is full!". Создаем метод get(), который возвращает объект, хранящийся в коробке, и очищает коробку. Внутри метода создаем временную переменную tempitem и присваиваем ей значение поля item. Обнуляем поле item, чтобы коробка стала пустой. Возвращаем значение tempitem. Создаем метод isEmpty(), который возвращает true, если коробка пуста, и false в противном случае.

## Тестирование

## 

## Код программы

https://github.com/SemenovIlya12/Jab/tree/Lab4

# Задание 2

## Текст задания

## Создайте ссылочный тип Сравнимое, гарантирующий наличие по данной ссылке метода со следующими характеристиками:

##  Называется “сравнить”

##  Принимает объект.

##  Тип принимаемого объекта может быть изменен без изменения самого Сравнимого.

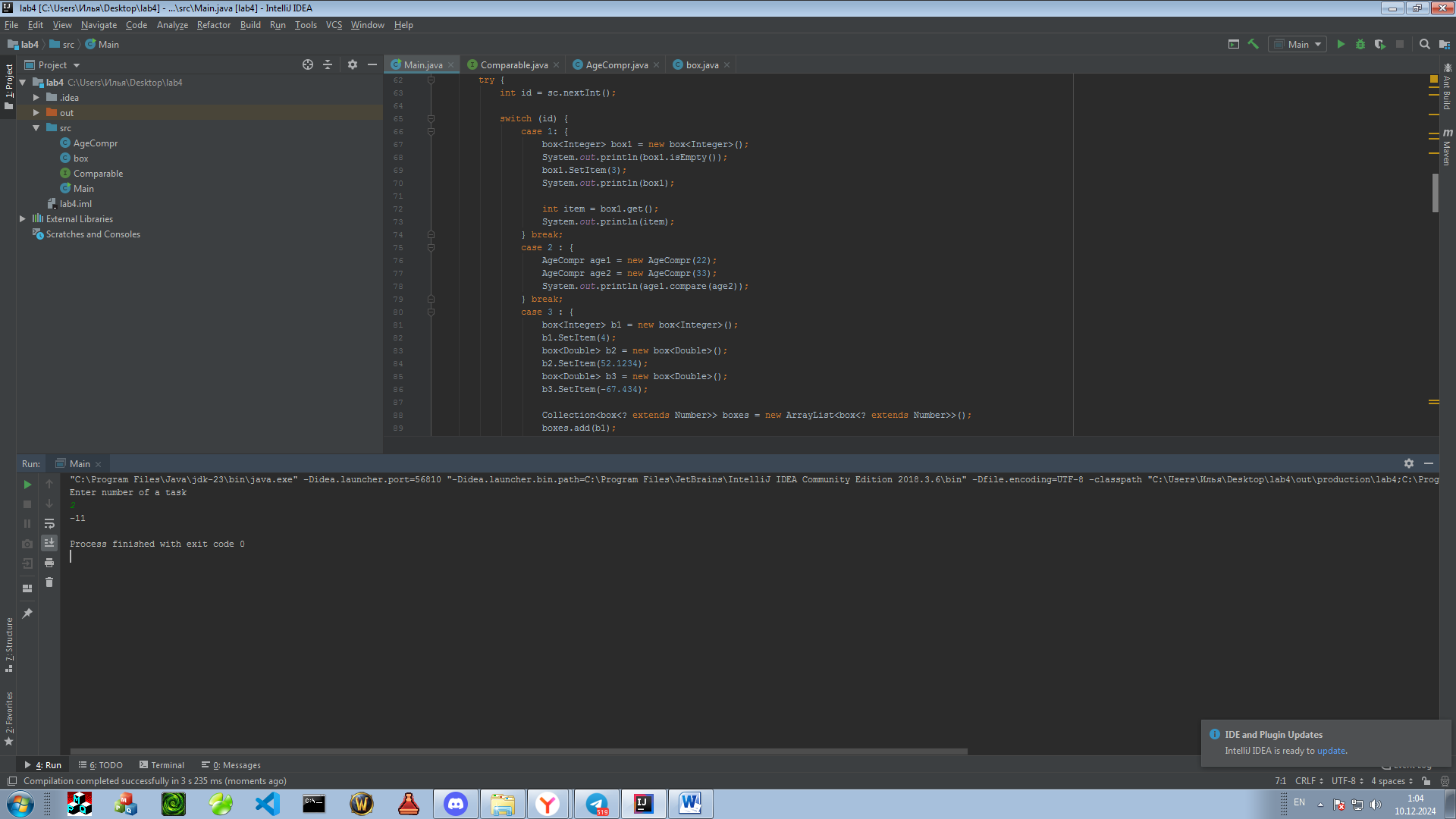
##  Возвращает целое число

## .Алгоритм решения

Создаем интерфейс Comparable<T>, где T — тип данных, с которым будет сравниваться объект.

В интерфейсе объявляем метод compare(T obj), который принимает объект типа T и возвращает целое число. Этот метод будет использоваться для сравнения текущего объекта с переданным объектом.

## Тестирование



## Код программы

Ссылка на githab

# Задание 3

Поиск максимума.

Создайте метод, принимающий набор Коробок из задачи 3.1.1 и возвращающий максимальное из их значений в формате double. Принимаемые методом Коробки могут быть параметризованы любыми видами чисел.

## Алгоритм решения

Создаем метод GetMaxInBoxes, который принимает коллекцию Collection<Box<? extends Number>> и возвращает максимальное значение из всех коробок в формате double.

В методе объявляем переменную max и инициализируем ее значением Double.NEGATIVE\_INFINITY, чтобы гарантировать, что любое число из коробок будет больше этого значения.

Используем цикл for-each для итерации по каждой коробке в коллекции boxes.

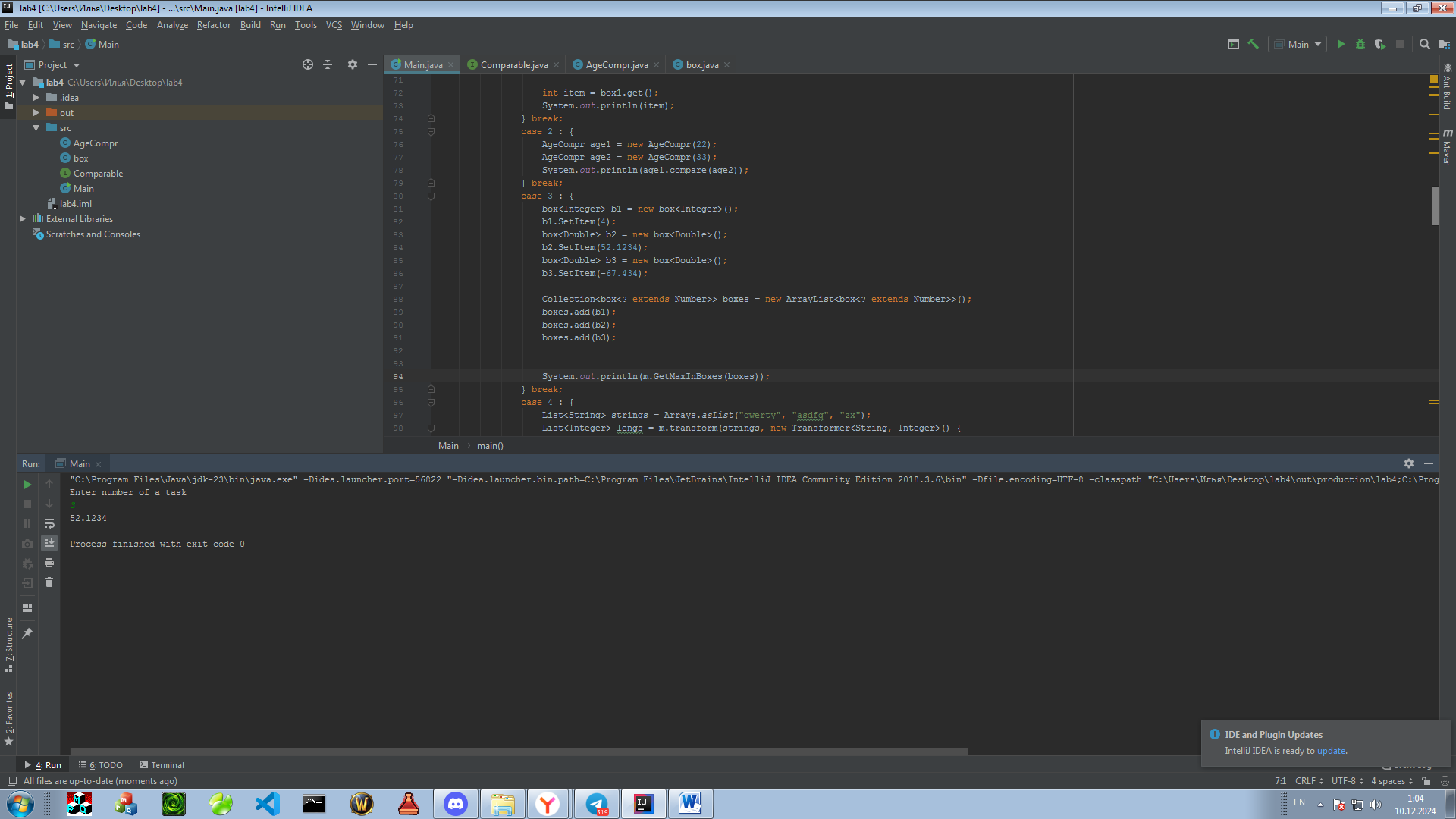
Для каждой коробки вызываем метод get(), чтобы извлечь объект типа Number, хранящийся в коробке.

Преобразуем полученный объект Number в тип double с помощью метода doubleValue().

Сравниваем полученное значение с текущим максимальным значением max. Если значение больше max, обновляем max.

После завершения итерации по всем коробкам, возвращаем значение max, которое теперь содержит максимальное значение из всех коробок.

## Тестирование



## Код программы

Ссылка на githab

# Задание 4

## Текст задания

## Функция.

## Разработайте такой метод, который будет принимать список значений типа T, и объект имеющий единственный метод apply. Данный метод надо применить к каждому элементу списка, и вернуть новый список значений типа P, при этом типы T и P могут совпадать, а могут не совпадать. Используйте разработанный метод следующим образом:

## 1. Передайте в метод список со значениями:“qwerty”, “asdfg”, “zx”, а получите список чисел, где каждое число соответствует длине каждой строки.

## 2. Передайте в метод список со значениями: 1,-3,7, а получите список в котором все отрицательные числа стали положительными, а положительные остались без изменений

## 3. Передайте в метод список состоящий из массивов целых чисел, а получите список в котором будут только максимальные значения каждого из исходных массивов

## Алгоритм решения

Создаем интерфейс Transformer<T, P>, где T — тип входных данных, а P — тип выходных данных.

В интерфейсе объявляем метод apply(T item), который принимает объект типа T и возвращает объект типа P. Этот метод будет использоваться для преобразования элементов списка.

Создаем метод transform, который принимает список значений типа T и объект типа Transformer<T, P>.

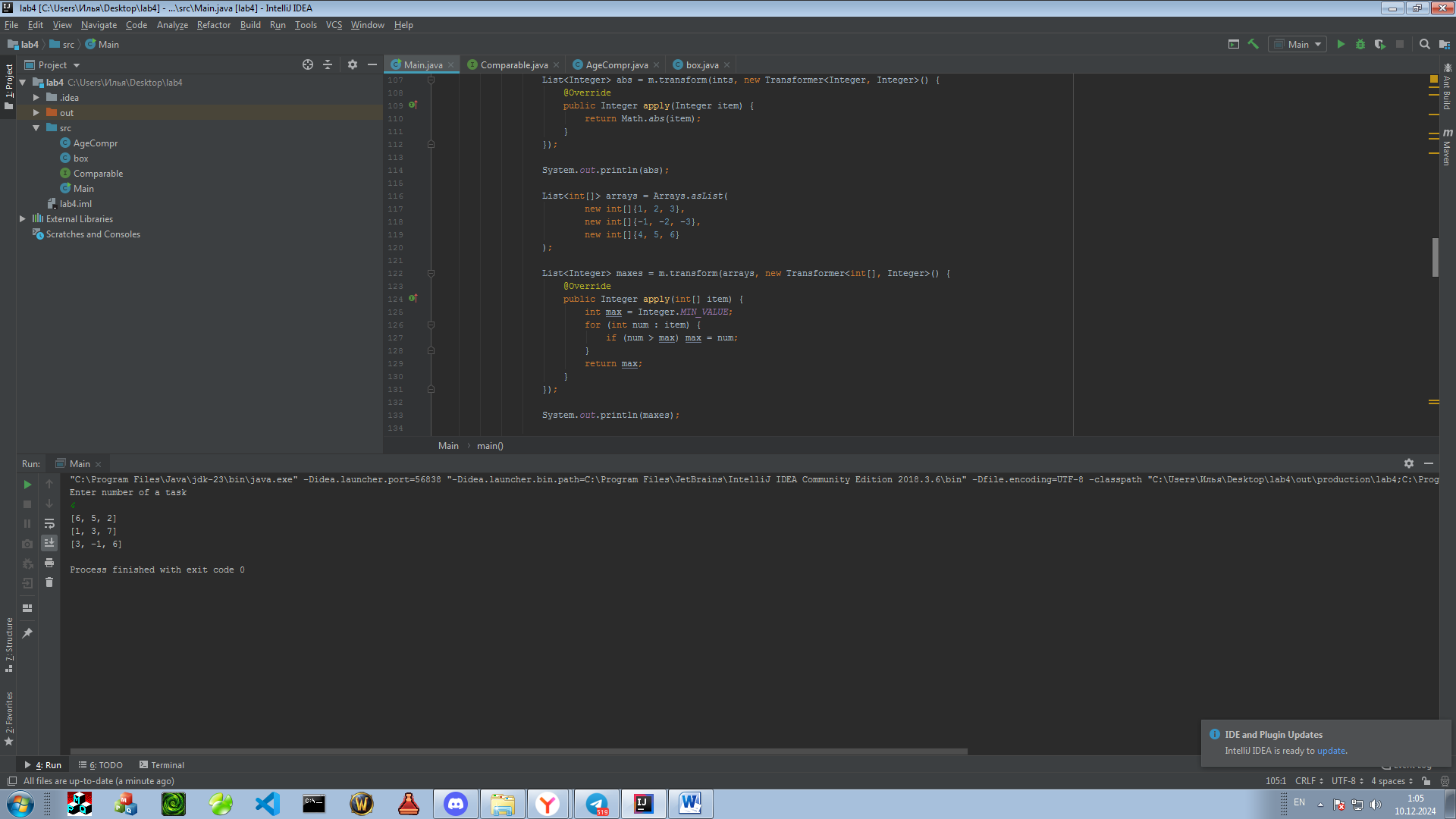
В методе создаем новый список result типа List<P>, который будет содержать преобразованные значения.

Используем цикл for-each для итерации по каждому элементу списка List.

Для каждого элемента вызываем метод apply объекта transformer, передавая ему текущий элемент. Результат вызова метода добавляем в список result.

После завершения итерации по всем элементам, возвращаем список result, который теперь содержит преобразованные значения.

## Тестирование



## Код программы

Ссылка на githab

# Задание 5

## Текст задания

Фильтр.

Разработайте такой метод, который будет принимать список значений типа T и объект имеющий

единственный метод test (принимает T и возвращает boolean). Верните новый список типа T, из которого удалены все значения не прошедшие проверку условием.

Используйте разработанный метод следующим образом:

1. Передайте в метод список со значениями: “qwerty”, “asdfg”, “zx”, и отфильтруйте все строки имеющие менее трех символов

2. Передайте в метод список со значениями: 1,-3,7, и отфильтруйте все положительные элементы

3. Передайте в метод список состоящий из массивов целых чисел, а получите список в котором будут только те массивы, в которых нет ни одного положительного элемента

## Алгоритм решения

Создаем интерфейс Tester<T>, где T — тип данных, который будет проверяться.

В интерфейсе объявляем метод Test(T item), который принимает объект типа T и возвращает boolean. Этот метод будет использоваться для проверки элементов списка.

Создаем метод TestList, который принимает список значений типа T и объект типа Tester<T>.

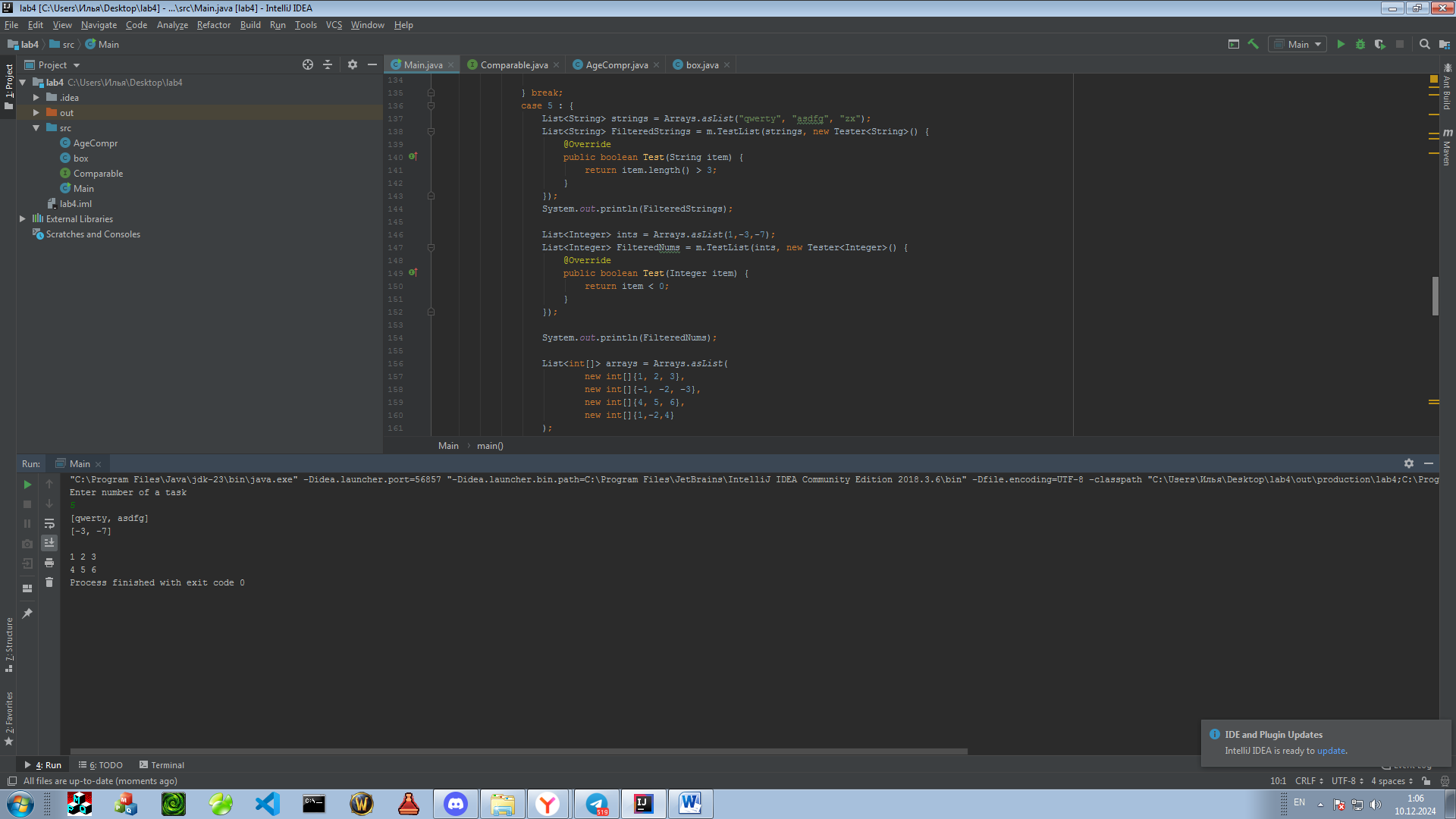
В методе создаем новый список result типа List<T>, который будет содержать отфильтрованные значения.

Используем цикл for-each для итерации по каждому элементу списка List.

Для каждого элемента вызываем метод Test , передавая ему текущий элемент. Если метод возвращает true, добавляем элемент в список result.

После завершения итерации по всем элементам, возвращаем список result, который теперь содержит только те элементы, которые прошли проверку.

## Тестирование



## Код программы

Ссылка на githab

# Задание 6

## Текст задания

## Сокращение.

## Разработайте такой метод, который будет принимать список значений типа T и способ с помощью которого список значений можно свести к одному значению типа T, которое и возвращается из метода.

## Используйте разработанный метод следующим образом:

## 1. Передайте в метод список со значениями: “qwerty”, “asdfg”, “zx”, и сформируйте одну большую строку, которая состоит из всех строк исходного списка.

## 2. Передайте в метод список со значениями: 1,-3,7, и верните сумму всех значений исходного списка.

## 3. Имеется список, состоящий из списков целых чисел, получите общеe количество элементов во всех списках. Подсказка: решить задачу можно в одно действие или последовательно использовать методы из 3.3.1 и 3.3.3.

## Далее необходимо изменить разработанный метод таким образом, чтобы данный методгарантированно не возвращал null и не выбрасывал ошибок в том случае, если исходный список пуст.

## Алгоритм решения

Создаем интерфейс ShrinkMethod<T>, где T — тип данных, который будет использоваться для сокращения списка.

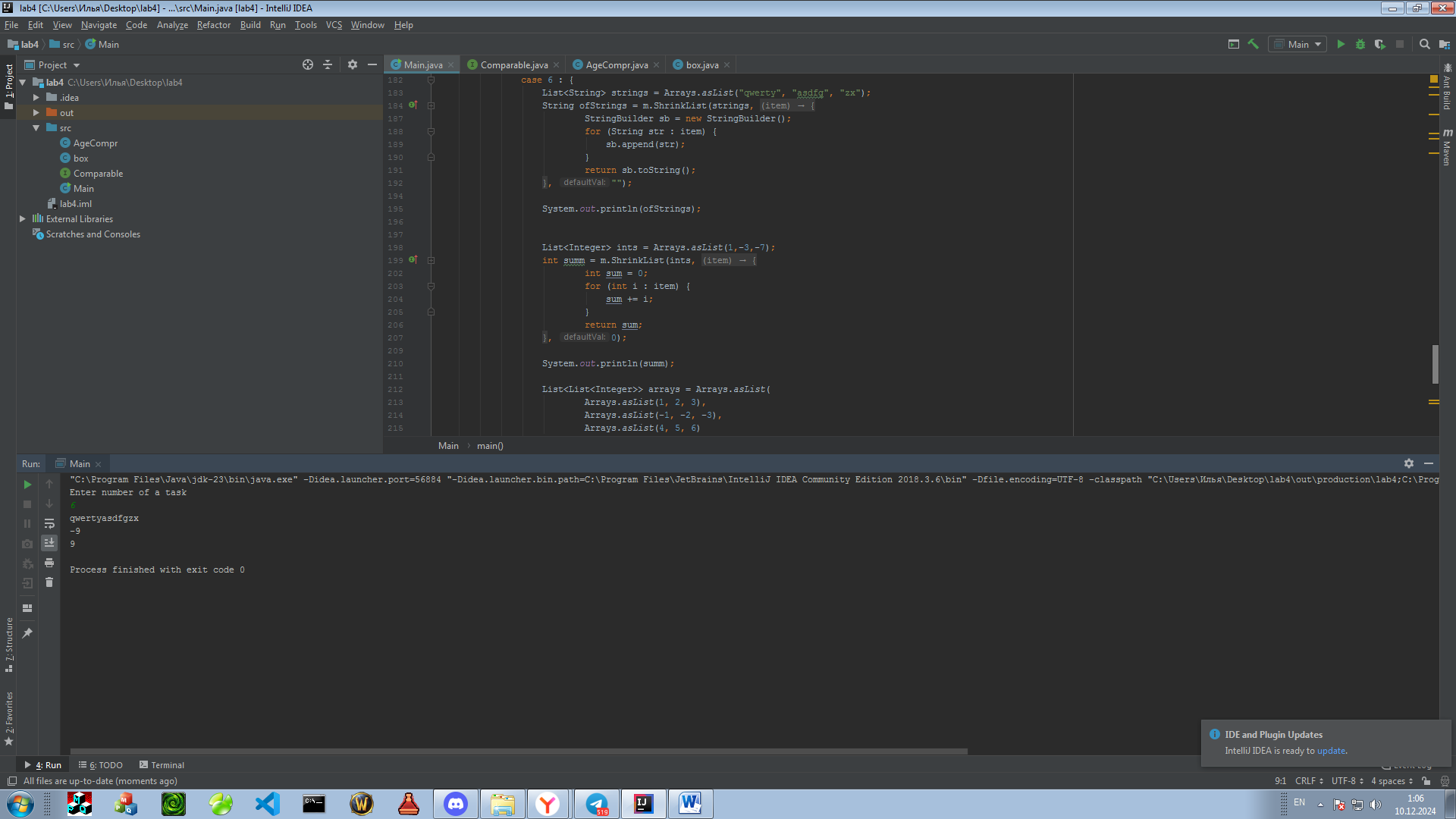
В интерфейсе объявляем метод Shrink(List<T> item), который принимает список значений типа T и возвращает одно значение типа T. Этот метод будет использоваться для сокращения списка до одного значения.

Создаем метод ShrinkList, который принимает список значений типа T, объект типа ShrinkMethod<T> и значение по умолчанию defaultVal.

В методе проверяем, является ли список List пустым. Если список пуст, возвращаем значение по умолчанию defaultVal.

Если список не пуст, вызываем метод Shrink объекта method, передавая ему список List, и возвращаем результат вызова метода.

## Тестирование



## Код программы

Ссылка на githab