**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное автономное образовательное учреждение высшего образования «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина

Институт радиоэлектроники и информационных технологий – РТФ

Центр ускоренного обучения

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

**По дисциплине «Программирование»**

Студент группы РИЗ-140938у Бабенко М.А.

Преподаватель: Архипов Н.А.

**Екатеринбург, 2024 г.**

**Лабораторная работа N3**

**Работа с операторами и одномерными массивами**

Цель: получить представление о работе с операторами и одномерными массивами в языке программирования Java.

**Задания для самостоятельной работы**

**Задание 1**. Напишите программу, в которой пользователь вводит целое число в диапазоне от 1 до 7, а программа определяет по этому числу день недели. Если введенное пользователем число выходит за допустимый диапазон, выводится сообщение о том, что введено некорректное значение. Используйте оператор выбора switch.

Число, вводимое с клавиатуры, записывается в целочисленную переменную **num**, значение которой проверяется при помощи оператора switch. Если число равно 1, в консоль выводится сообщение о том, что это – понедельник; если 2 – вторник, и т.д, Варианты кода между собой разграничены оператором break, без которого будут выполняться все последующие условия подряд. Условие, когда число не входит в заданный диапазон, содержится в варианте «default», в котором выводится сообщение об ошибке. Результат работы программы и ее код показан на рисунке 1.

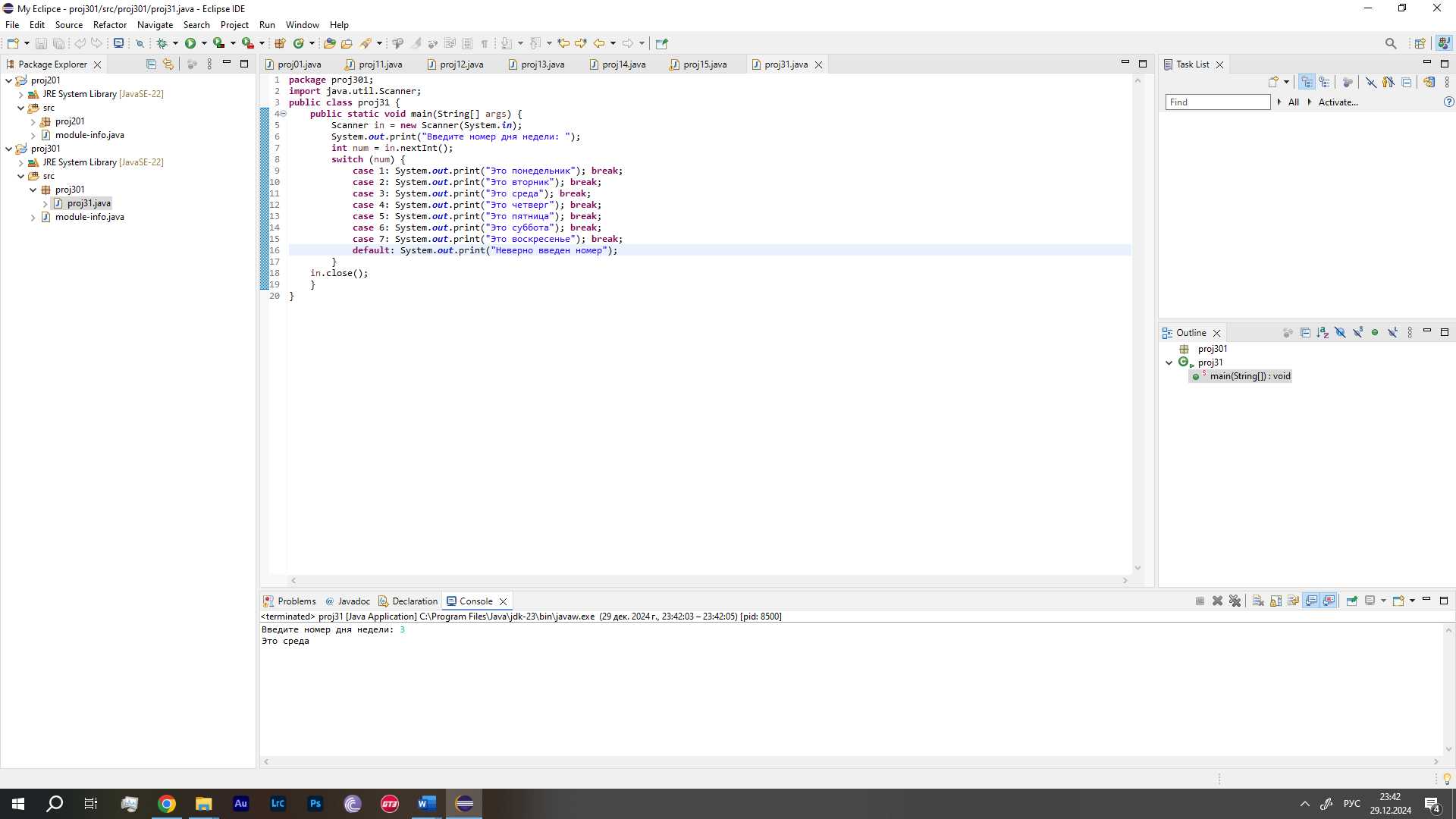


Рисунок 1 – Код и результат работы программы proj31

**Задание 2**. Напишите программу, в которой пользователю предлагается ввести название дня недели. По введенному названию программа определяет порядковый номер дня в неделе. Если пользователь вводит неправильное название дня, программа выводит сообщение о том, что такого дня нет. Предложите версию программы на основе вложенных условных операторов и на основе оператора выбора switch.

Название (пишется с заглавной буквы), вводимое с клавиатуры, записывается в строку **day**, значение которой проверяется при помощи оператора switch (поддерживается с Java версии 7). Если значение строки равно «Понедельник», в консоль выводится сообщение о том, что это – день под номером 1; если вторник –2, и т.д, Варианты кода между собой разграничены оператором break, без которого будут выполняться все последующие условия подряд. Условие, когда число не входит в заданный диапазон, содержится в варианте «default», в котором выводится сообщение об ошибке. Результат работы программы и ее код показан на рисунке 2.

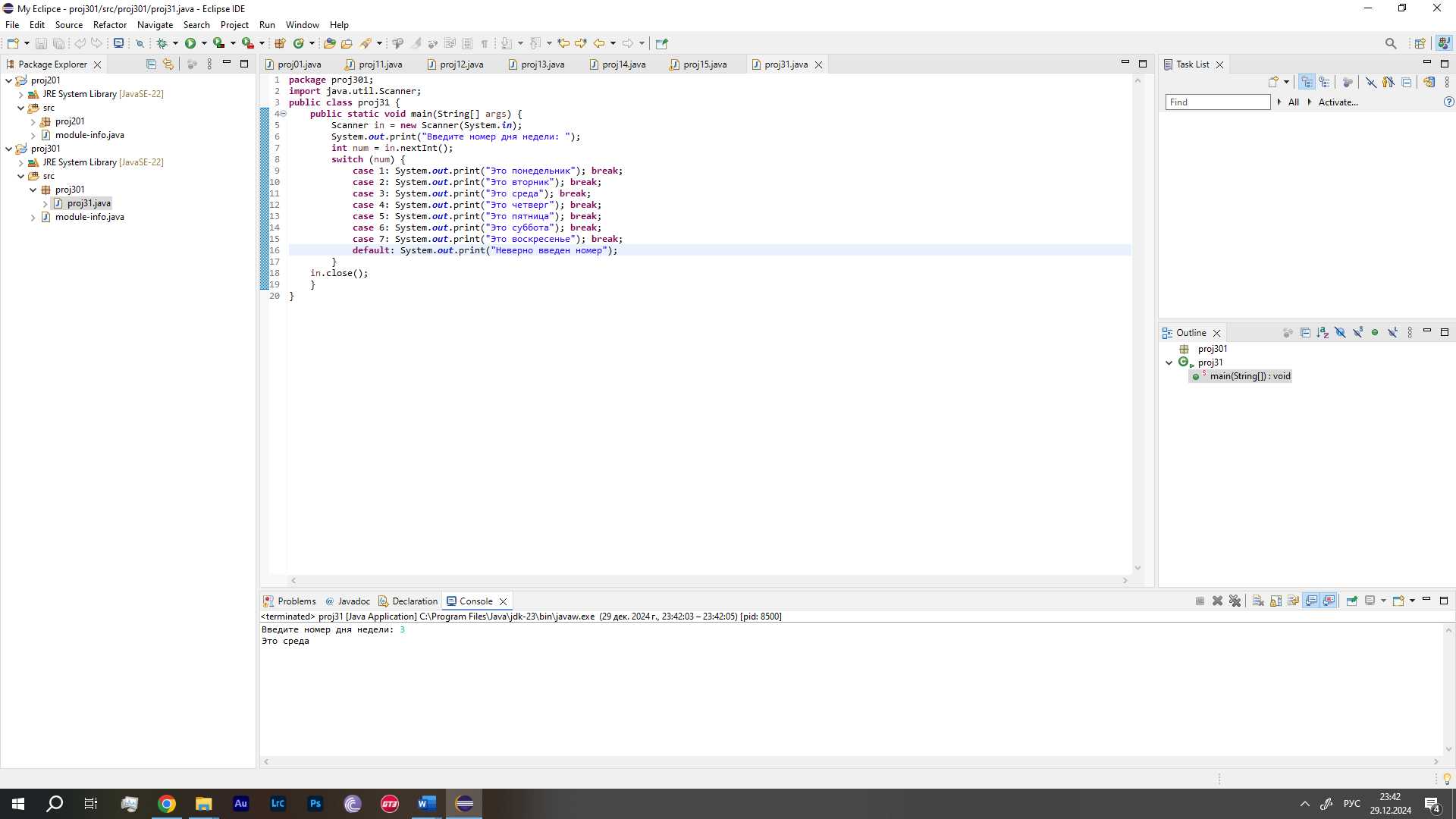


Рисунок 2 – Код и результат работы программы proj32

**Задание 3**. Напишите программу, которая выводит последовательность чисел Фибоначчи. Первые два числа в этой последовательности равны 1, а каждое следующее число равно сумме двух предыдущих (получается последовательность 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89 и так далее). Количество чисел в последовательности вводится пользователем. Предложите версии программы, использующие разные операторы цикла.

Пользователь вводит в консоль количество выводимых чисел. Затем, программой проверяется, чтобы количество чисел не было равно нулю. В этом случае программой выводится соответствующее сообщение («Число должно быть больше 0!»), и программа досрочно завершается **с ошибкой**, код которой равен 1 (неуспешное завершение программы). Для первых двух чисел (равным 1 или 2), выводятся заранее сгенерированные значения. Затем, программой проверяется значение переменной **num**, т.е. количество генерируемых чисел. Если оно равно 1 или 2, выводятся одно или два первых числа соответственно, и программа завершается аналогичным образом. Если количество чисел равно трем и больше, программой генерируются и остальные числа. Первый вариант программы был написан с использованием оператора цикла **for**. Результат ошибки во время работы программы (было введено число 0) и ее код показан на рисунке 3.

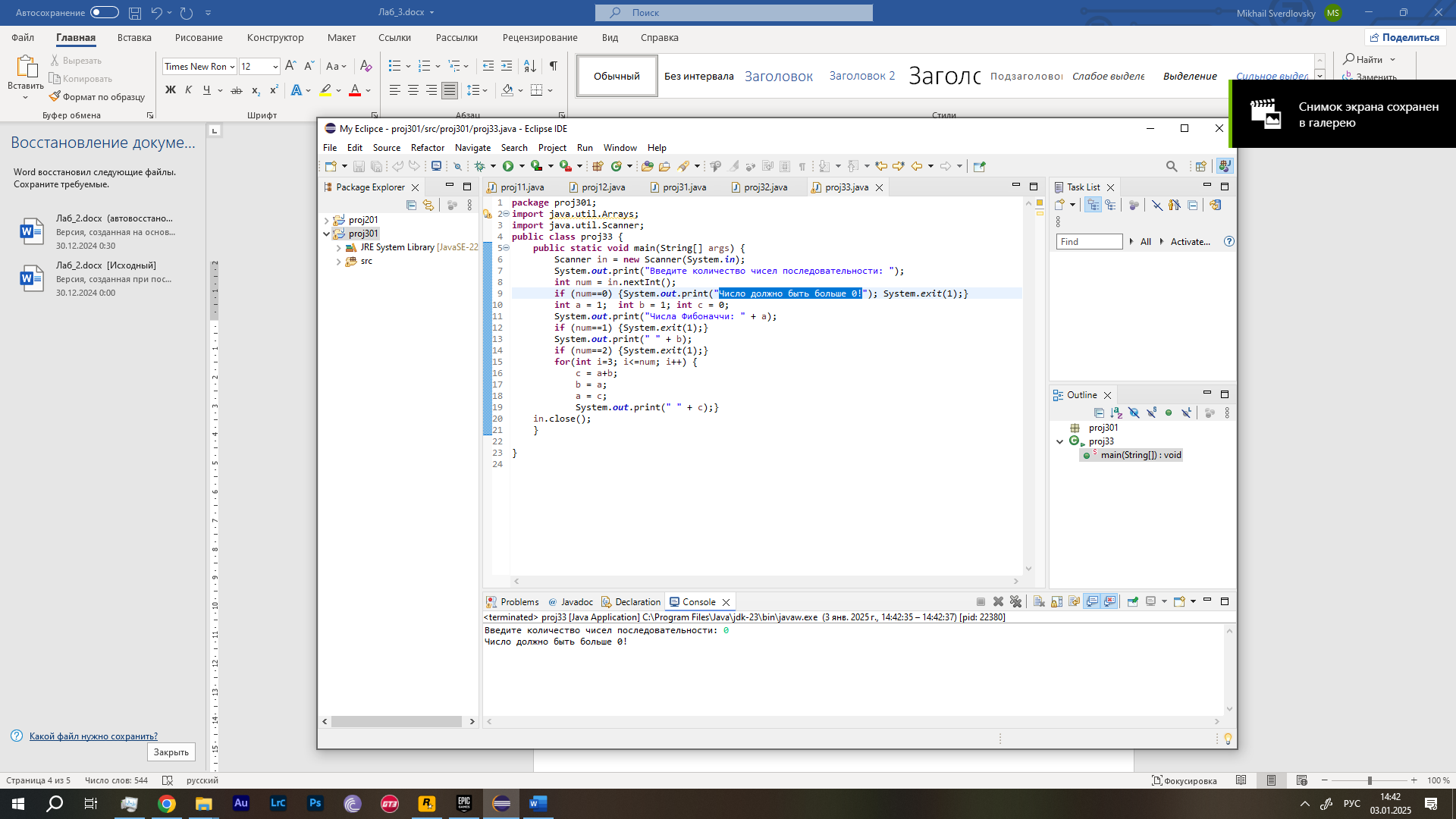


Рисунок 3 – Результат ошибки во время работы программы proj33

Результат успешной работы программы и ее код показан на рисунке 4.

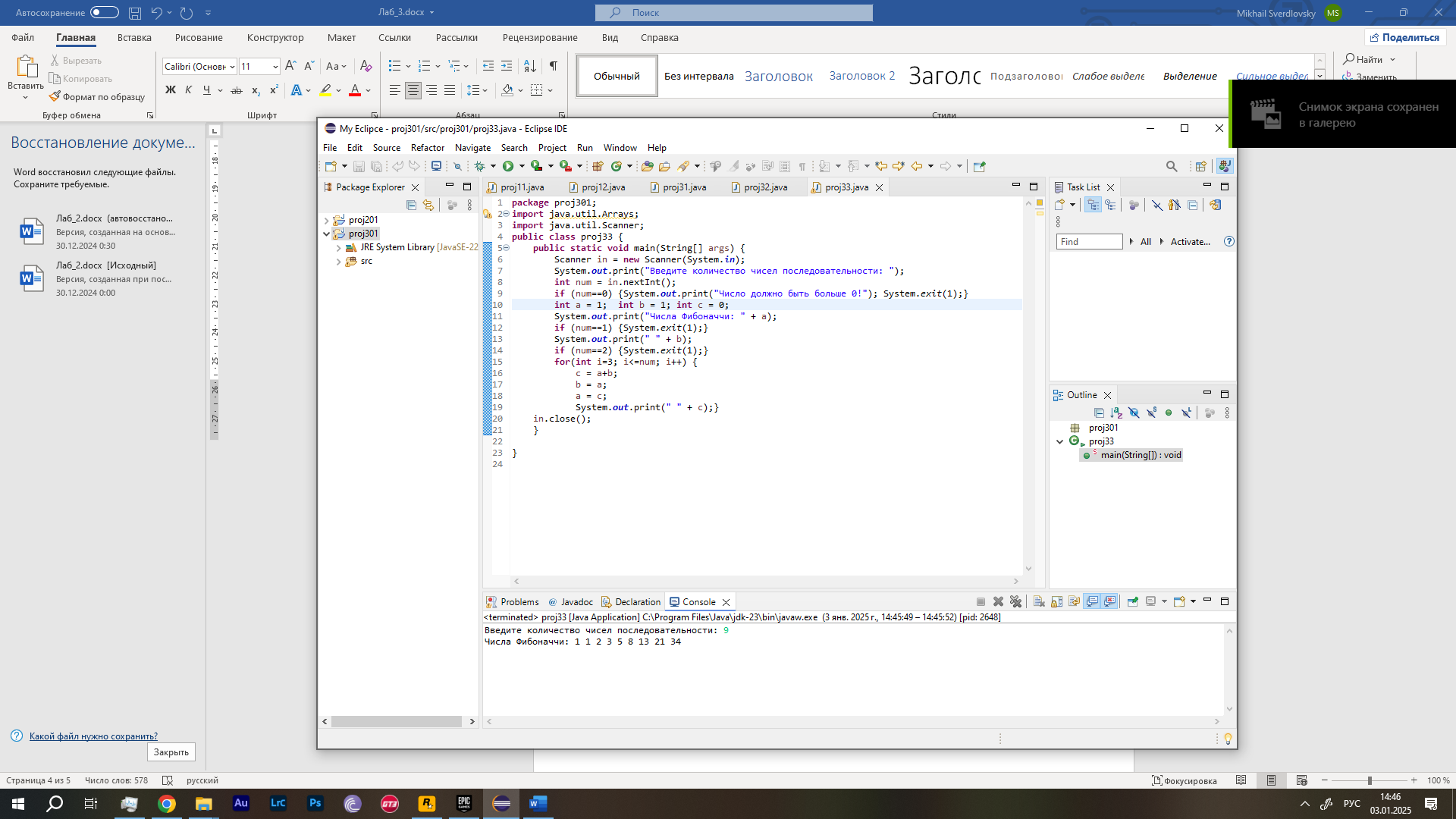


Рисунок 4 – Код и результат успешной работы программы proj33

Второй вариант программы был написан с использованием оператора цикла с предусловием **while**. Результат успешной работы программы и ее код показан на рисунке 5.

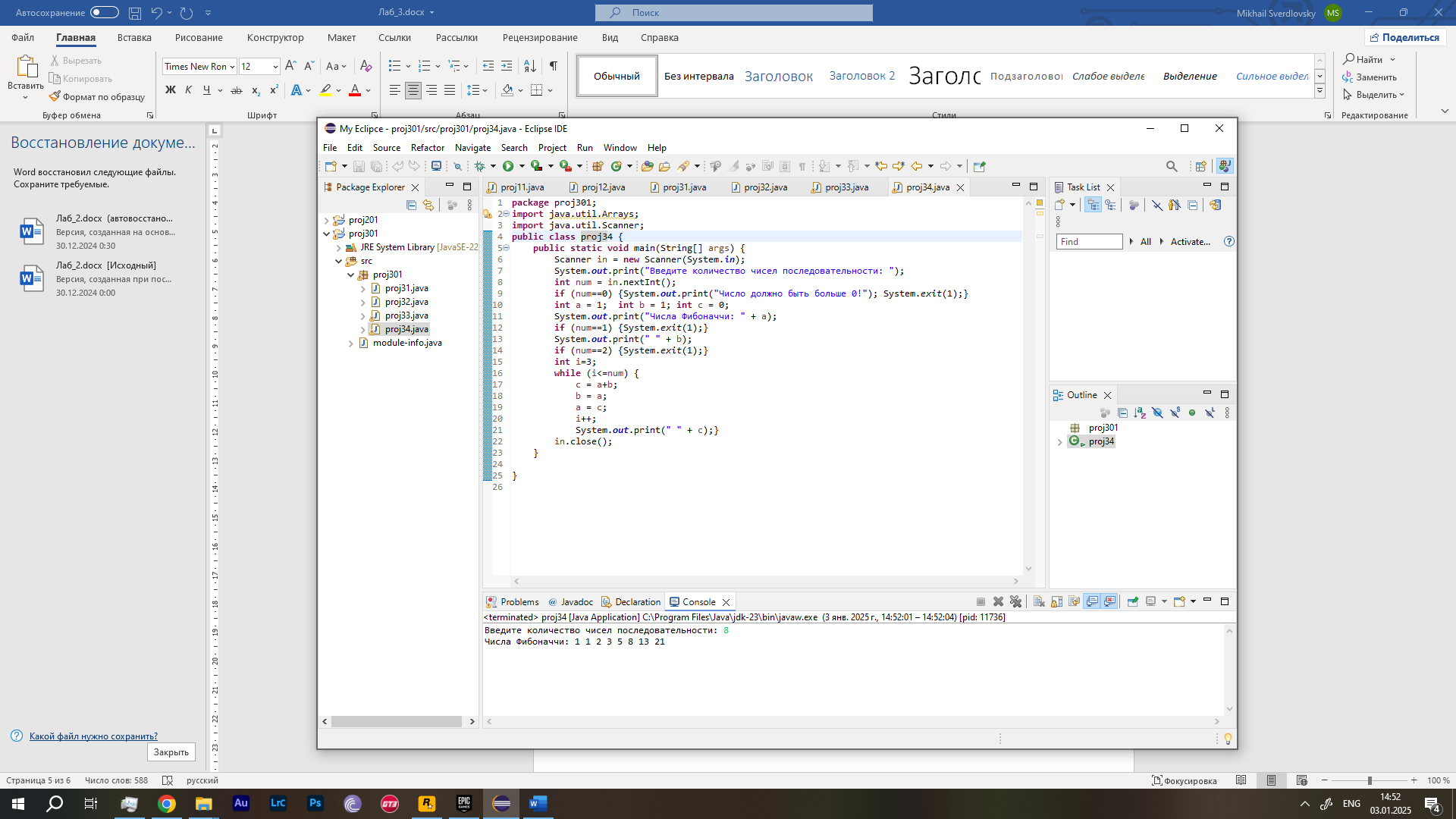


Рисунок 5 – Код и результат успешной работы программы proj34

Третий вариант программы был написан с использованием оператора цикла с постусловием **do…while**. Результат успешной работы программы и ее код показан на рисунке 6.

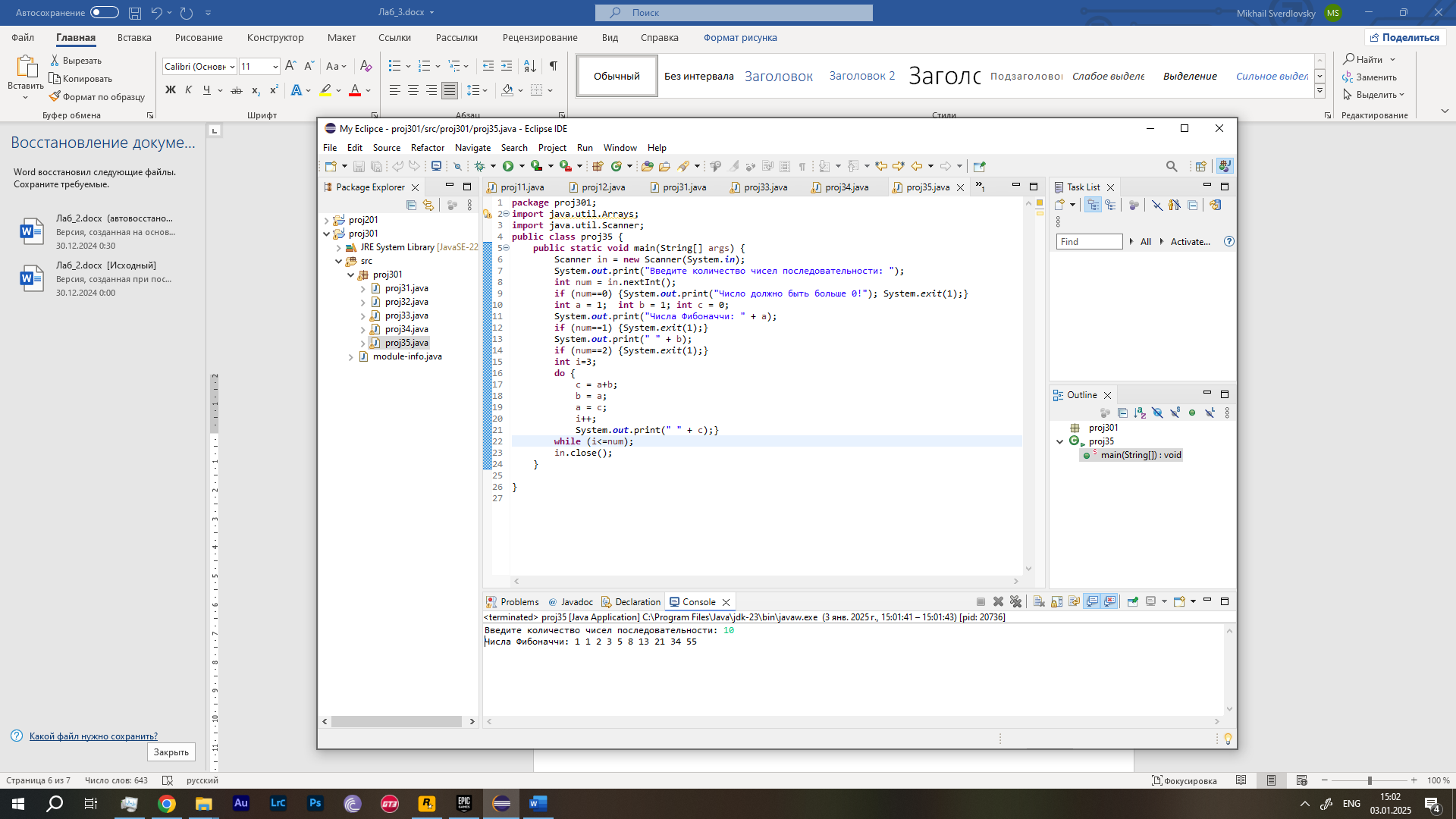


Рисунок 6 – Код и результат успешной работы программы proj35

**Задание 4**. Напишите программу, в которой пользователем вводится два целых числа. Программа выводит все целые числа — начиная с наименьшего (из двух введенных чисел) и заканчивая наибольшим (из двух введенных чисел). Предложите разные версии программы (с использованием разных операторов цикла).

Для данной программы определяется переменная **num** – количество вводимых чисел массива. Также мы заранее должны подключить пакет **java.util.Arrays**, для сортировки чисел внутри массива. Затем, в коде программы определяется массив **mass**, длинна которого равна переменной **num**. Используя цикл **for**, производим пошаговую запись чисел в массив; после чего цикл завершается, и функцией **Arrays.sort(mass)** (где mass – название нашего массива), производится его сортировка по возрастанию. Затем, для вывода элементов массива, мы также используем цикл **for**. Результат успешной работы программы и ее код показан на рисунке 7.

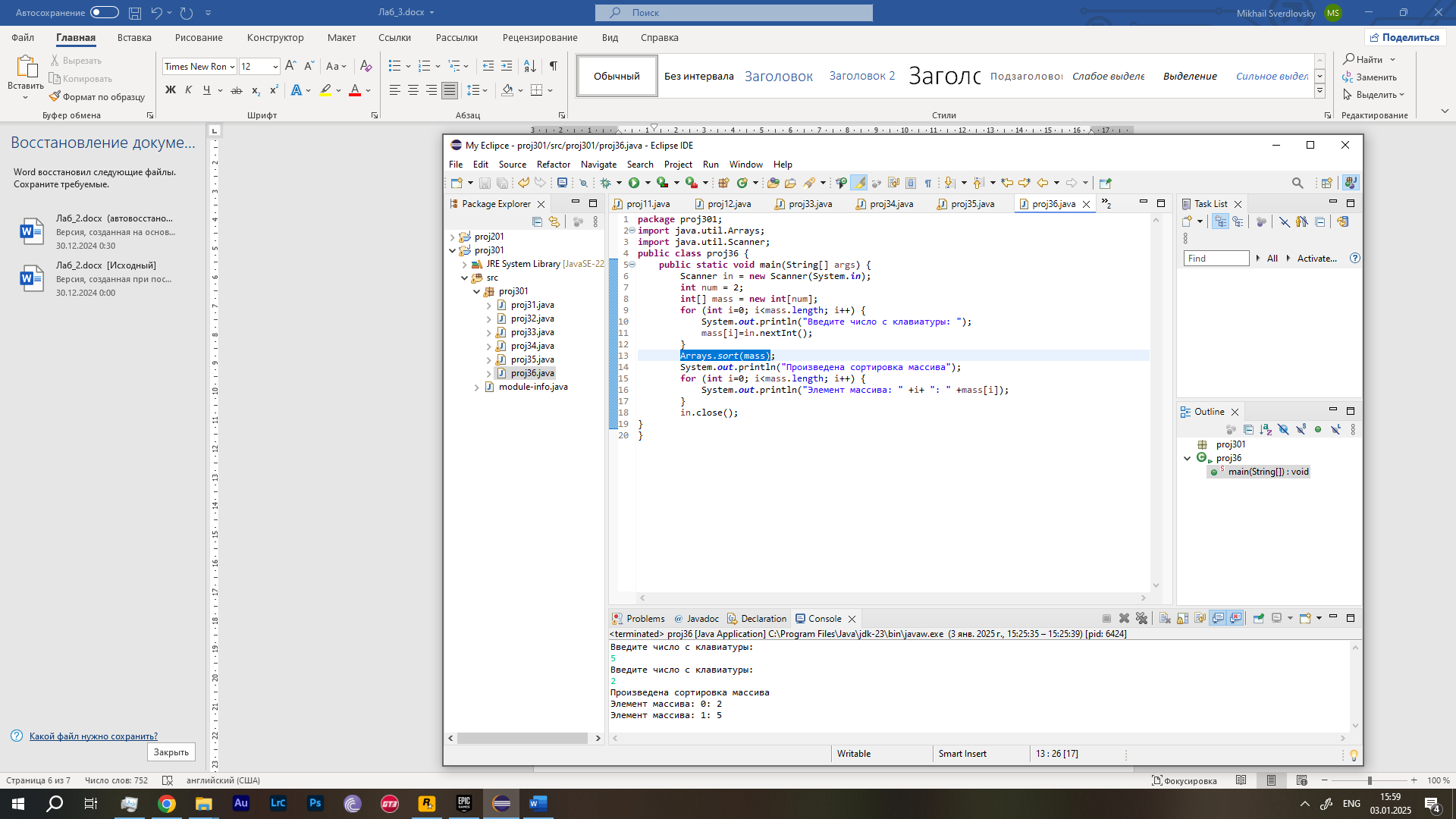


Рисунок 7 – Код и результат успешной работы программы proj36

Второй вариант программы был написан с использованием функций цикла **while** и **do…while**. Результат работы программы и ее код показан на рисунке 8.

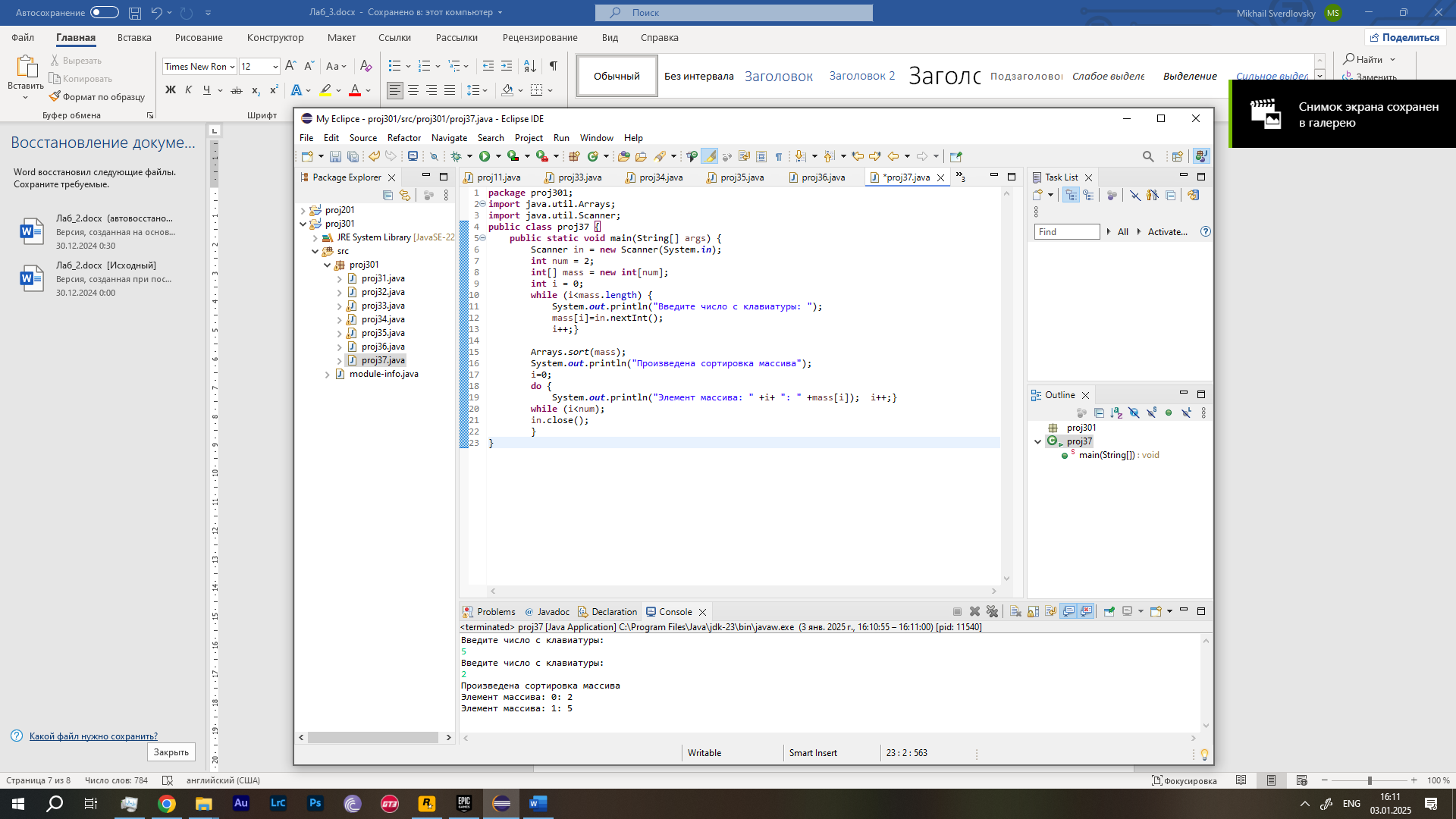


Рисунок 8 – Код и результат успешной работы программы proj37

**Задание 5**. Напишите программу, в которой вычисляется сумма чисел, удовлетворяющих таким критериям: при делении числа на 5 в остатке получается 2, или при делении на 3 в остатке получается 1 Количество чисел в сумме вводится пользователем. Программа отображает числа, которые суммируются, и значение суммы. Предложите версии программы, использующие разные операторы цикла.

Количество суммируемых чисел (чисел, которые удовлетворяют условиям), вводится пользователем с клавиатуры, и записывается в переменную **num**; количество должно быть больше 0, в противном случае будет выведено сообщение об ошибке, и программа завершит свою работу. Затем, командой объявляется массив **mass**, длинна которого равна значению **num**. Числа в данный массив записываются за счет использования функции цикла **for**, в котором также имеется функция цикла **while**, при помощи которой происходит поиск соответствующих условиям чисел (если значение переменной **a** не делится на 5 с остатком 2 или не делится на 3 с остатком 1, проверяется следующее число; в противном случае число записывается в i-тый элемент массива). Затем, за счет использования функции цикла **for**, происходит поэлементный вывод массива на экран. После вывода, переменная **a** обнуляется и в дальнейшем используется для расчета суммы всех элементов массива. Результат успешной работы программы и ее код показан на рисунке 9.

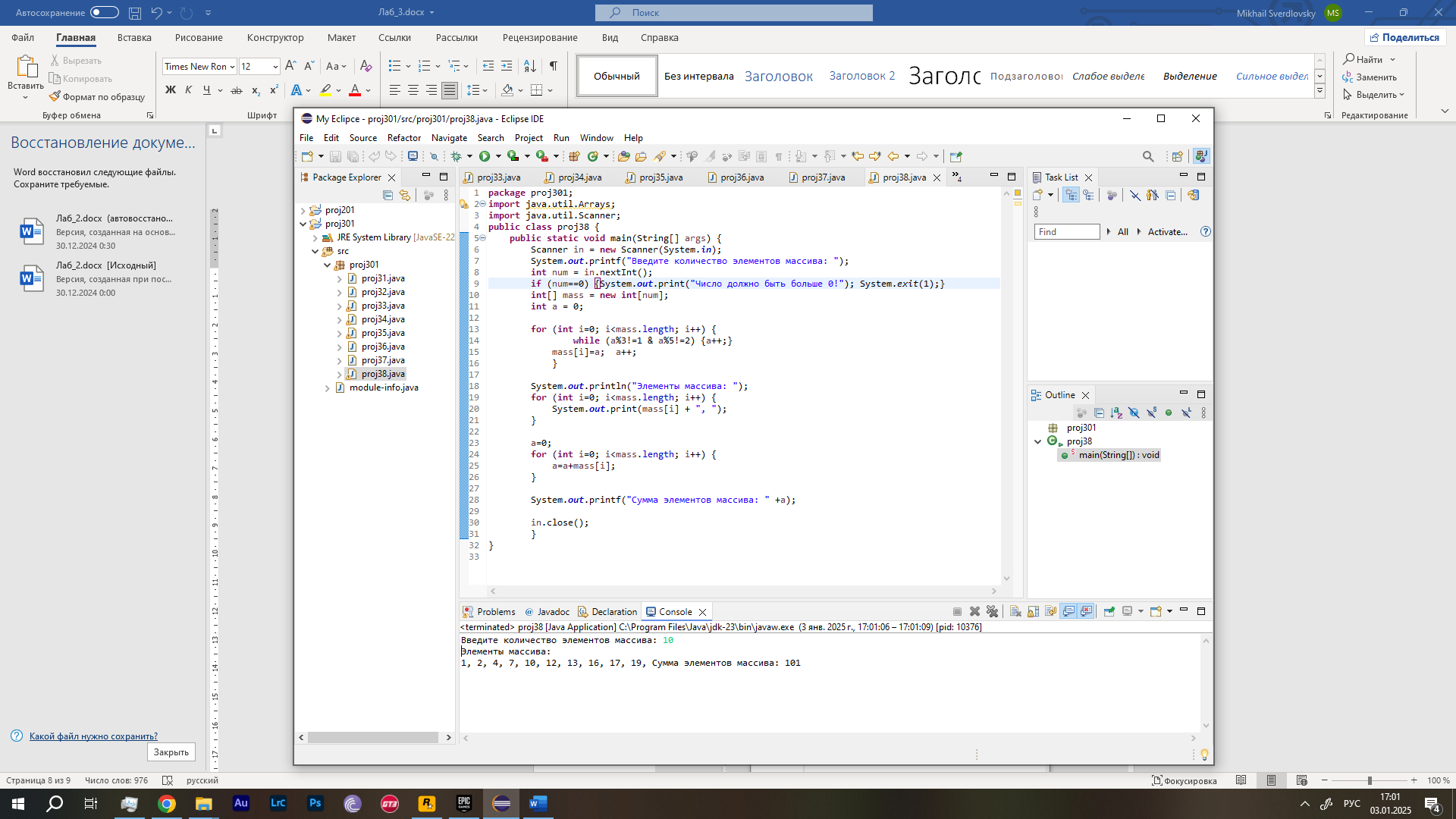


Рисунок 9 – Код и результат успешной работы программы proj38

Второй вариант программы был написан с использованием функций цикла **while** вместо функции **for**. Результат работы программы и ее код показан на рисунке 10.

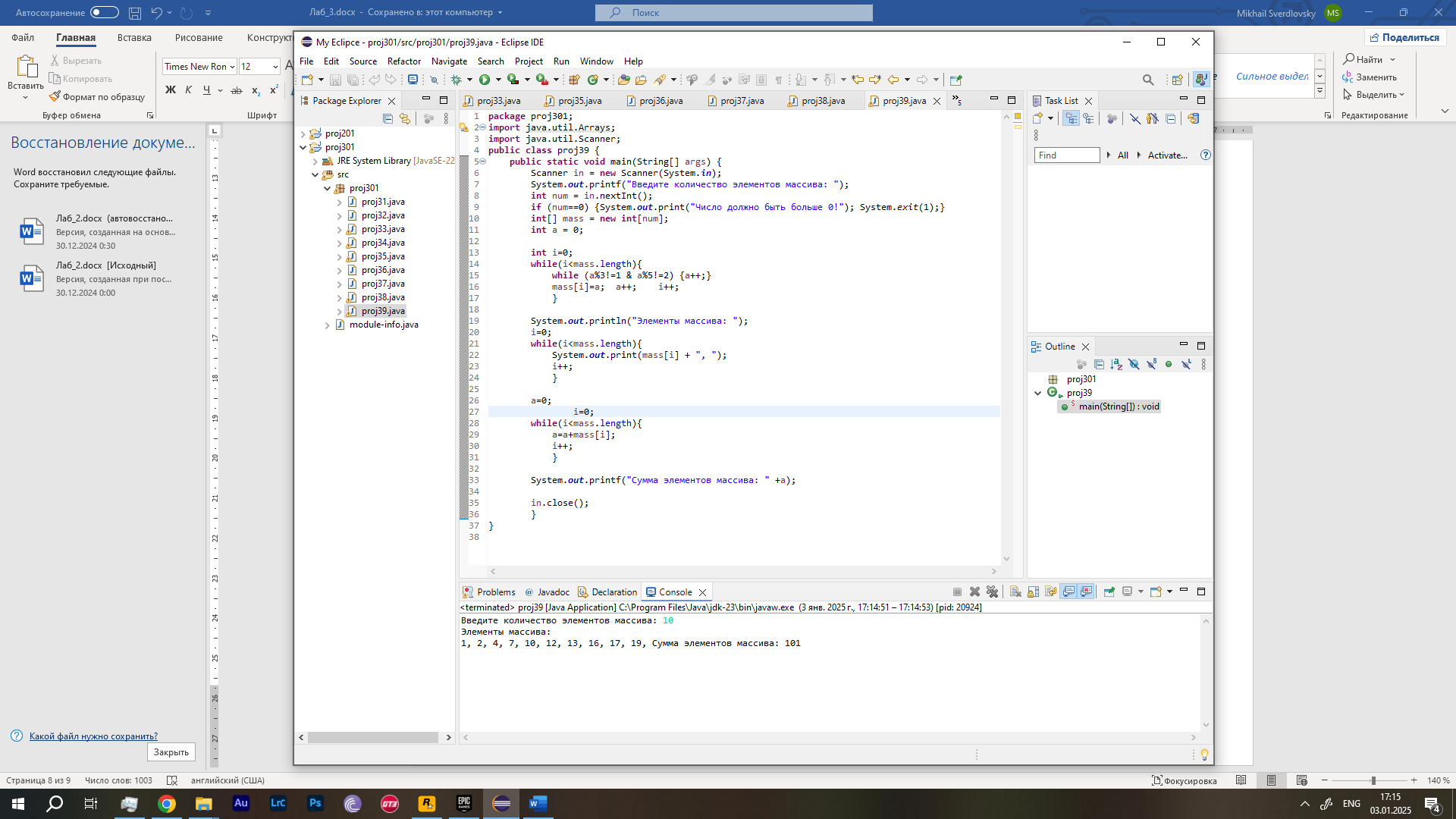


Рисунок 10 – Код и результат успешной работы программы proj39

**Задание 6**. Напишите программу, в которой создается одномерный числовой массив и заполняется числами, которые при делении на 5 дают в остатке 2 (числа 2, 7,12,17 и так далее). Размер массива вводится пользователем. Предусмотреть обработку ошибки, связанной с вводом некорректного значения.

Количество суммируемых чисел (чисел, которые удовлетворяют условиям), вводится пользователем с клавиатуры, и записывается в переменную **num**; количество должно быть больше 0, в противном случае будет выведено сообщение об ошибке, и программа завершит свою работу. Затем, командой объявляется массив **mass**, длинна которого равна значению **num**. Числа в данный массив записываются за счет использования функции цикла **for**, в котором также имеется функция цикла **while**, при помощи которой происходит поиск соответствующих условиям чисел (если значение переменной **a** не делится на 5 с остатком 2, проверяется следующее число; в противном случае число записывается в i-тый элемент массива). Затем, за счет использования функции цикла **for**, происходит поэлементный вывод массива на экран. Результат успешной работы программы и ее код показан на рисунке 11.

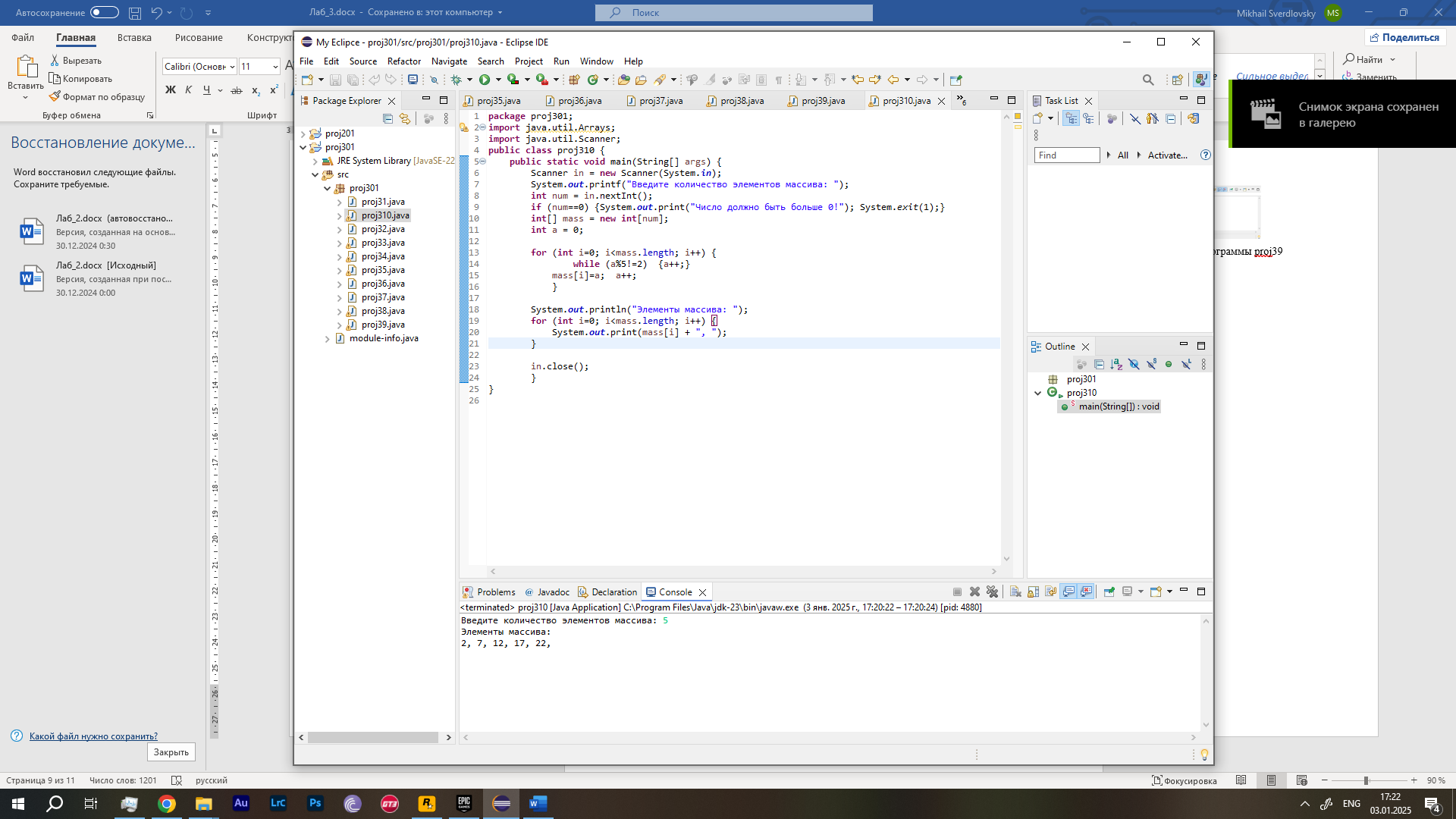


Рисунок 11 – Код и результат успешной работы программы proj310

Результат ошибки во время работы программы (было введено число 0) показан на рисунке 12.

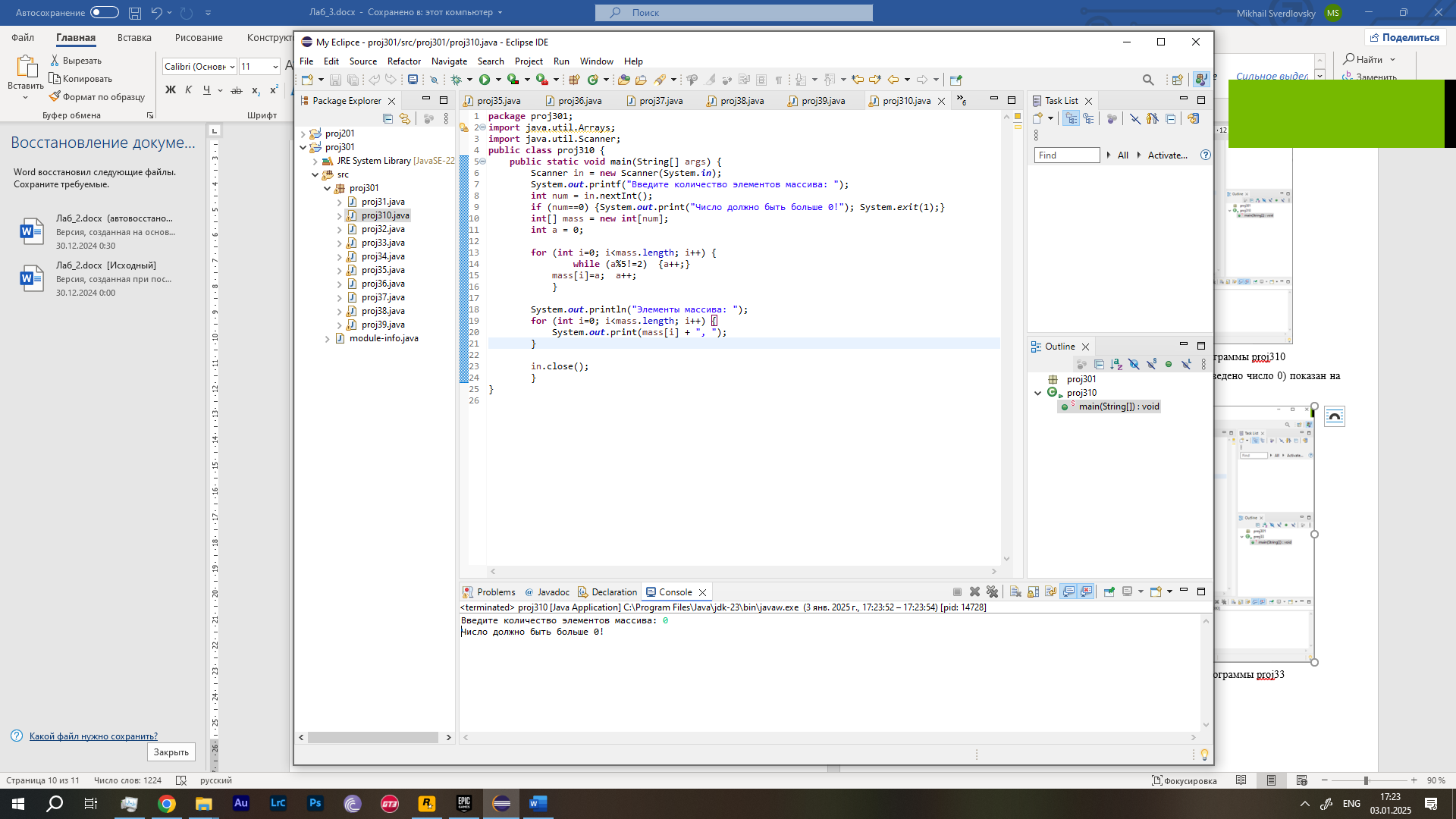


Рисунок 12 – Результат ошибки во время работы программы proj310

**Задание 7**. Напишите программу, в которой создается одномерный символьный массив из 10 элементов. Массив заполняется буквами «через одну», начиная с буквы ' а ': то есть массив заполняется буквами ' а ' , ' с ' , ' е ' , ' д ' и так далее. Отобразите массив в консольном окне в прямом и обратном порядке. Размер массива задается переменной.

Размер создаваемого массива определяется значением переменной **int=10**. Для записи букв используем переменную **a** типа **char**, т.к. этот тип данных поддерживает инкрементирование. Сама запись в массив реализуется при помощи оператора цикла **for**, в котором используем двойное инкрементирование. Прямой вывод массива осуществляется, когда в цикле переменная i равна 0, а по мере прохождения циклов она инкрементируется. Вывод массива в обратном порядке осуществляется, когда переменная **i равна 9** (до тех пор пока она не будет больше или равна нулю), а по мере прохождения циклов она декрементируется. Результат работы программы и ее код показан на рисунке 13.

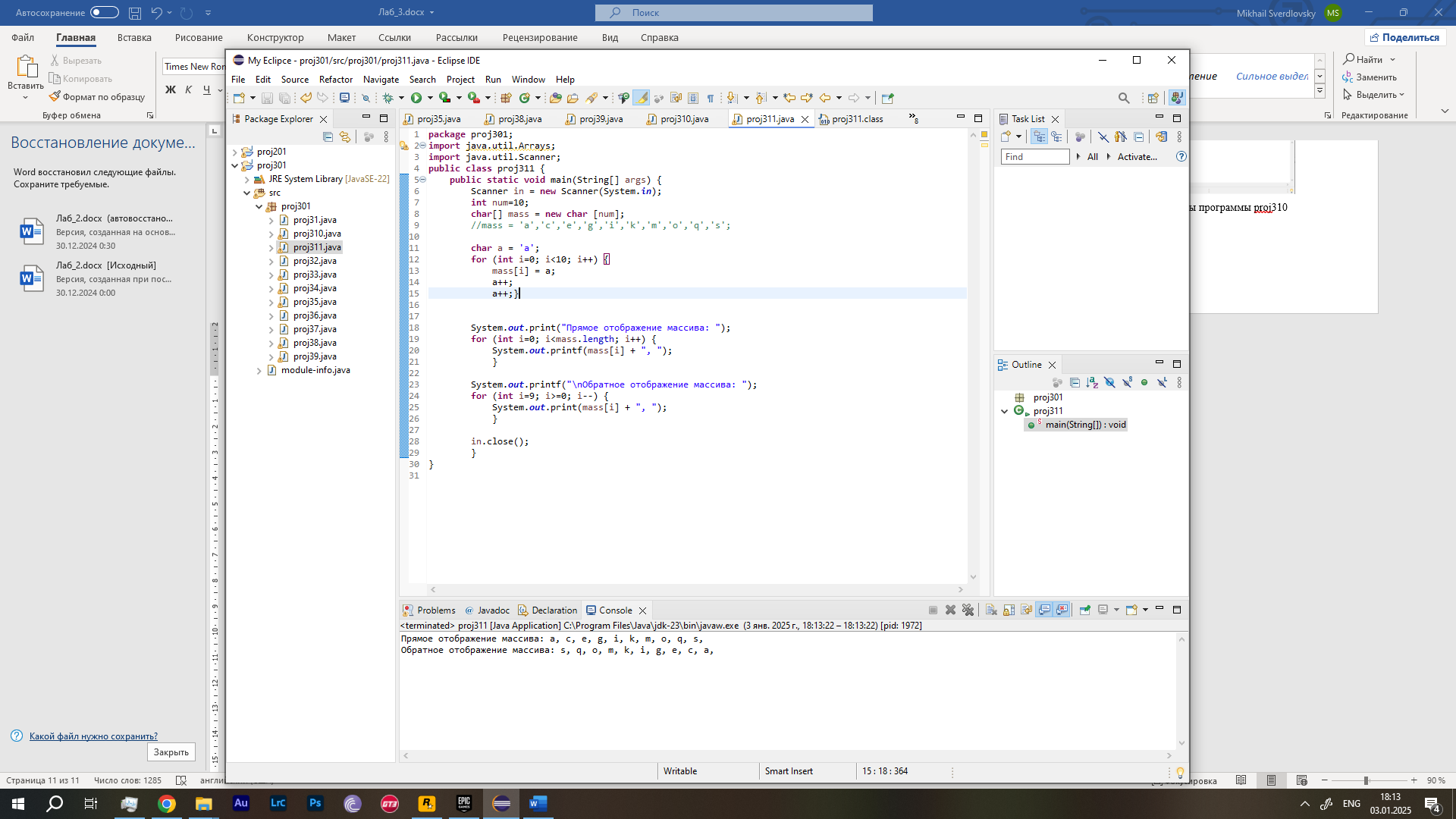


Рисунок 13 – Код и результат работы программы proj311

**Задание 8**. Напишите программу, в которой создается символьный массив из 10 элементов. Массив заполнить большими (прописными) буквами английского алфавита. Буквы берутся подряд, но только согласные (то есть гласные буквы ’ А ' , ‘ Е ' и ' I ' при присваивании значений элементам массива нужно пропустить). Отобразите содержимое созданного массива в консольном окне.

Размер создаваемого массива определяется значением переменной **int=10**. Для записи букв используем переменную **a** типа **char**. Сама запись в массив реализуется при помощи оператора цикла **while**, в котором используем инкрементирование счетчика **i** и значения переменной **a**. В этот цикл вкладывается оператор условия **if**, позволяющий исключить запись гласных букв (путем дополнительного инкрементирования переменной **a**). Результат работы программы и ее код показан на рисунке 14.

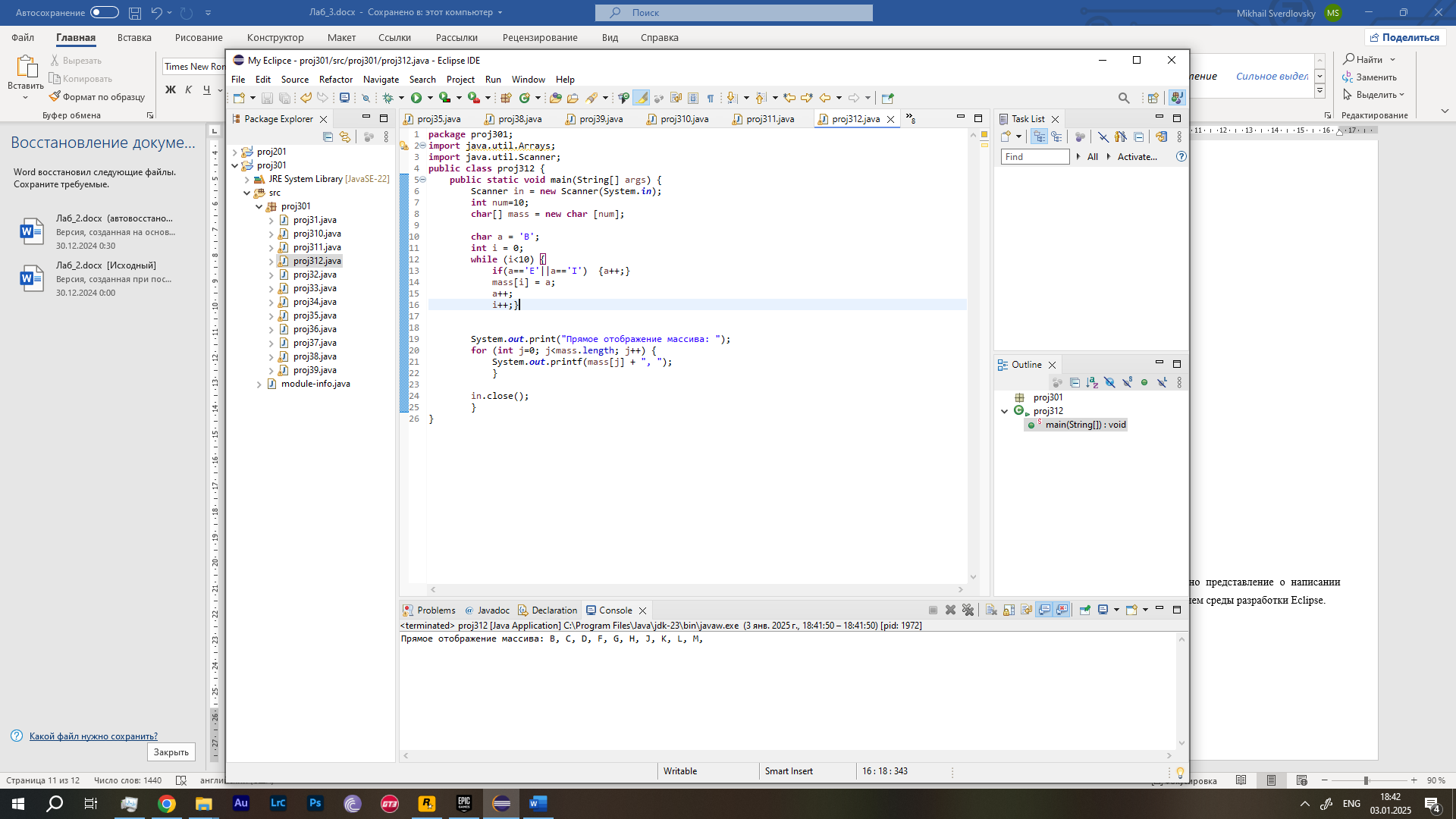


Рисунок 14 – Код и результат работы программы proj312

**Задание 9**. Напишите программу, в которой создается массив п заполняется случайными числами. Массив отображается в консольном окне. В этом массиве необходимо определить элемент с минимальным значением. В частности, программа должна вывести значение элемента с минимальным значением и индекс этого элемента. Если элементов с минимальным значением несколько, должны быть выведены индексы всех этих элементов.

Размер создаваемого массива определяется целочисленной переменной **num**. В случае если значение переменной равно 0, будет выведено сообщение об ошибке, и программа завершится. Для использовании функции генерации случайных чисел, следует подключить пакет **import java.util.Random**. Для обращения к элементам массива мы используем оператор цикла **for;** а в коде **random.nextInt(200),** 200 – диапазон генерации случайных чисел. Затем, используя оператор цикла for, производим вывод всех элементов массива, после чего вводим две переменные: **min** – для записи наименьшего значения массива (по умолчанию присваивается значение нулевого элемента массива) и **mini** – для записи индекса наименьшего значения массива (по умолчанию присваивается значение индекса нулевого элемента массива). Сравнение элементов массива производятся при помощи оператора **if**, вложенного в оператор цикла **for**. Результат работы программы и ее код показан на рисунке 15.

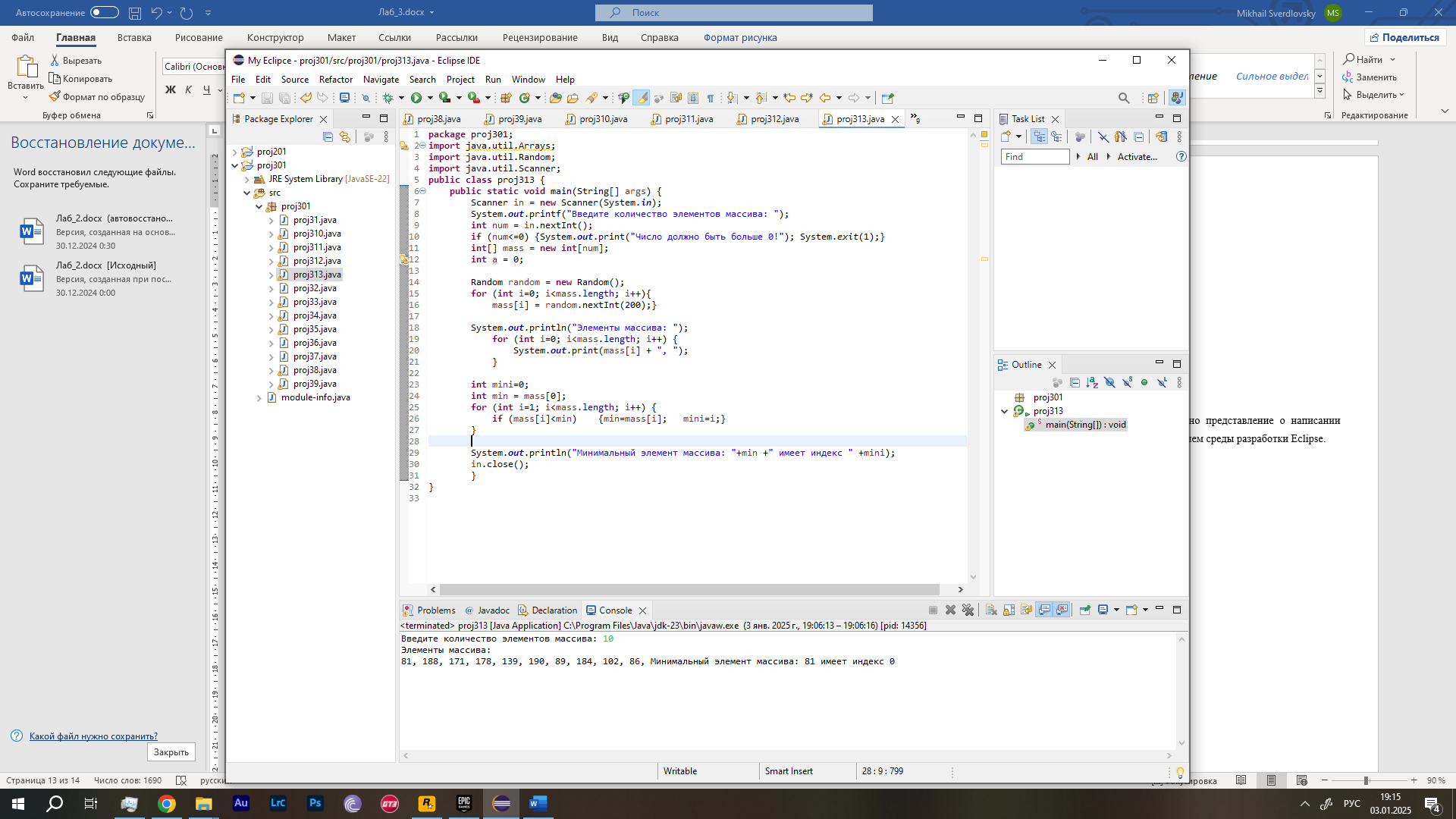


Рисунок 15 – Код и результат работы программы proj313

**Задание 10**. Напишите программу, в которой создается целочисленный массив, заполняется случайными числами и после этого значения элементов в массиве сортируются в порядке убывания значений.

Количество чисел в массиве вводится пользователем с клавиатуры, и записывается в переменную **num**; количество должно быть больше 0, в противном случае будет выведено сообщение об ошибке, и программа завершит свою работу. Затем, командой объявляется массив **mass**, длинна которого равна значению **num**. Числа в данный массив записываются за счет использования функции цикла **for**, путем генерации этих самых чисел (**random.nextInt(200))**. После происходит сортировка массива по возрастанию, за это отвечает функция **Arrays.sort(mass)**. После сортировки массива по возрастанию, вводим переменные для хранения временных значений: **tmp, m, j**. Чтобы отсортировать массив по убыванию, меняем местами уже отсортированные элементы массива (последний – с первым, предпоследний – во вторым; и т.д.). Затем, с помощью функции **for**, в консоль выводится уже отсортированный по убыванию массив **mass**. Результат работы программы и ее код показан на рисунке 16.

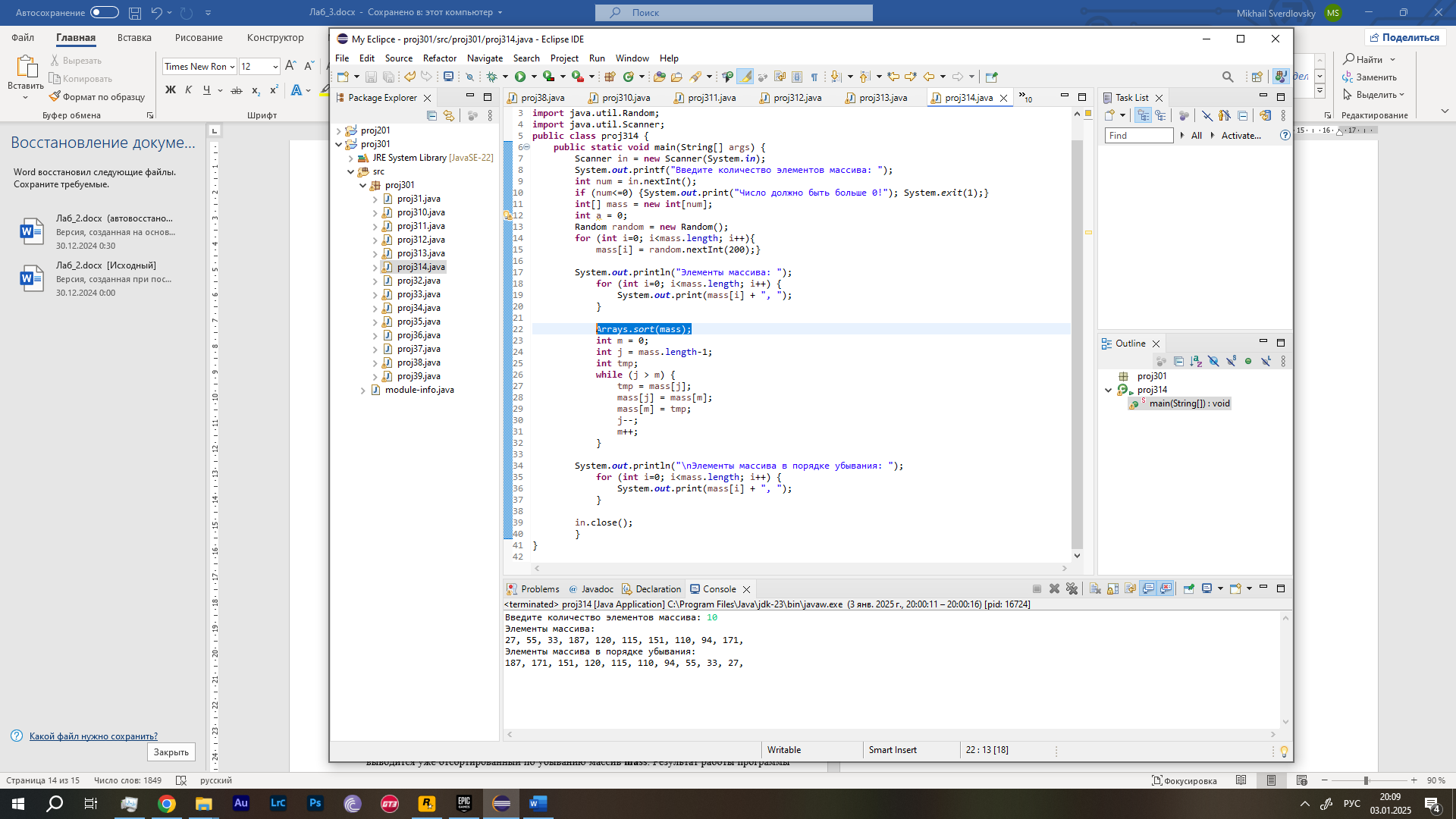


Рисунок 16 – Код и результат успешной работы программы proj314

**Задания из Timus**

**Задание 1306.** Пусть задана последовательность из N целых неотрицательных чисел. Медианой такой последовательности в случае нечетного N называется элемент, который будет равноудален от концов последовательности, если ее отсортировать по возрастанию или убыванию (нетрудно сообразить, что этот элемент имеет номер (N+1)/2 в отсортированной последовательности, если номера считать с единицы). В случае четного N медианой называется среднее арифметическое двух элементов, которые окажутся на местах N/2 и (N/2)+1, если последовательность отсортировать. Однако исходная последовательность не обязана быть отсортированной. Напишите программу, которая по заданной входной последовательности вычисляет ее медиану.

Пользователь вводит в консоли значение целочисленной переменной n – количество чисел. Затем, создается целочисленный массив nums, длинной n, и пользователь по очереди вводит числа в этот массив. Когда все данные введены, происходит сортировка массива по возрастанию, это потребуется в дальнейшем для вычисления медианы. Затем вводится переменная типа double и происходит расчет медианного числа. Полученное значение выводится в консоль.

Результат работы программы и ее код показан на рисунке 17.

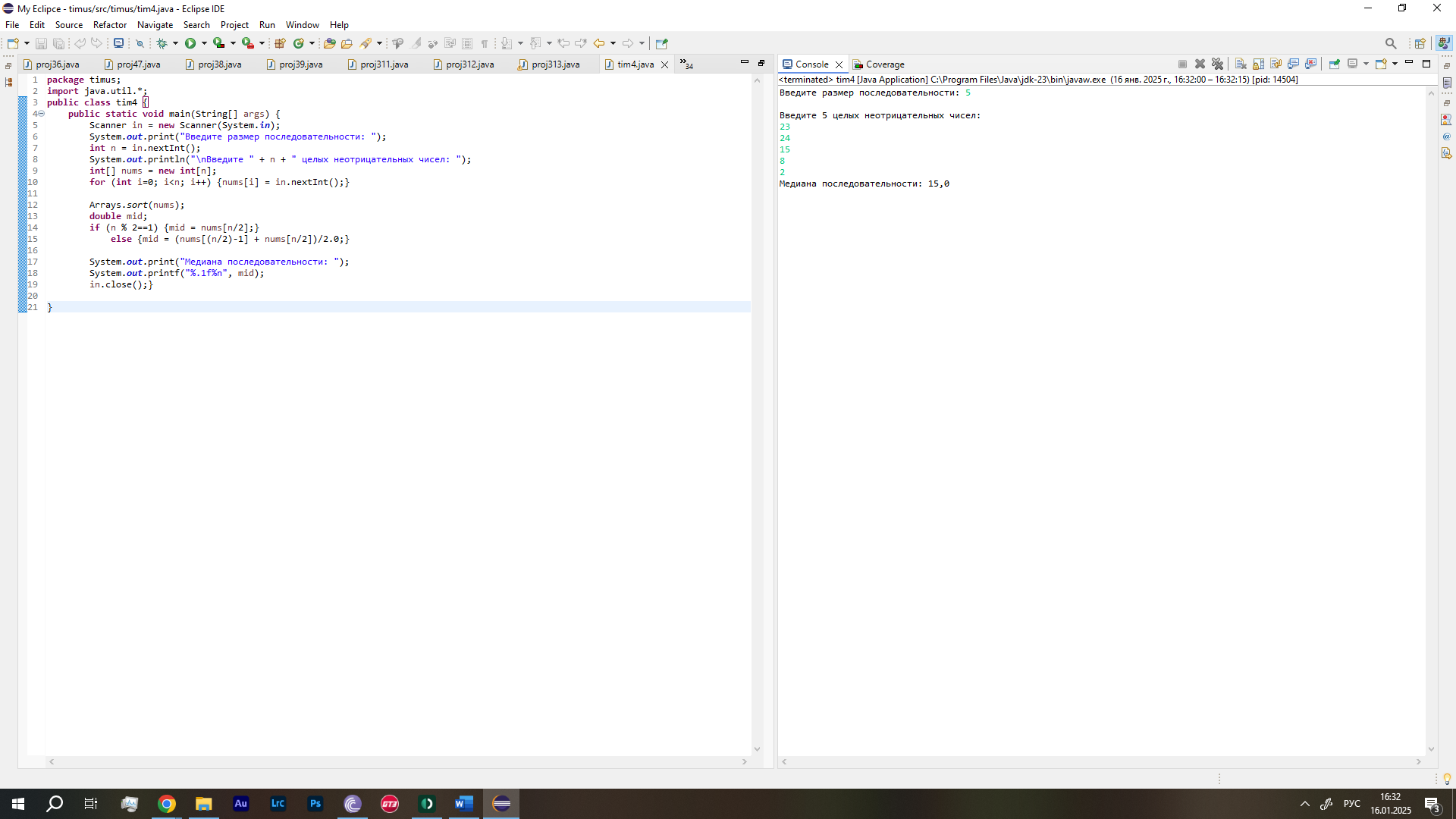


Рисунок 17 – Код и результат работы программы tim4

**Задание 1306.** В университете следующая политика предоставления стипендии:

а) если у студента есть тройки, то стипендия ему не выплачивается;

б) если студент сдал сессию на одни пятерки, то он получает именную стипендию;

В) если студент не получил именную стипендию, и его средний балл не менее 4.5, то он получает повышенную стипендию;

г) если студент не получил ни именную, ни повышенную стипендии, и при этом у него нет троек, то он получает обычную стипендию;

Помогите Васе определить, будет ли у него стипендия, и если да, то какая.

Пользователем вводится в целочисленную переменную n количество экзаменов (оценок) полученных за семестр, а в массив ngrade – полученные оценки. Переменная three получает значение 1

Результат работы программы и ее код показан на рисунке 17.



Рисунок 17 – Код и результат работы программы tim5

**Вывод**

В результате лабораторной работы было получено представление о работе с операторами и одномерными массивами в языке программирования Java.