**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 ЗАДАНИЯМ С САЙТА TIMUS**

**По дисциплине «Программирование»**

Студент группы РИЗ-140938у Бабенко М.А.

Преподаватель: Архипов Н.А.

**Екатеринбург, 2025 г.**

**Решение заданий с сайта Tumus**

**Ссылка на репозиторий GitHub:** https://github.com/sverdlovsky00/Tumus\_LW

**Задание 1079**. Напишите программу, которая для заданного значения n находит максимальное среди чисел a0, a1, …, an.

Программа вычисляет максимум в последовательности чисел 𝑎i для каждого заданного 𝑛, где последовательность определяется следующим образом: для четных a[i/2], для нечетных a[i/2]+a[i/2+1]. Все значения, используемые в расчетах заданы изначально и записаны в int[] inputArray. Для каждого n из входных данных программа проходит массив sequence от находит максимальное значение. Результат работы программы и ее программный код представлен на рисунке 1.

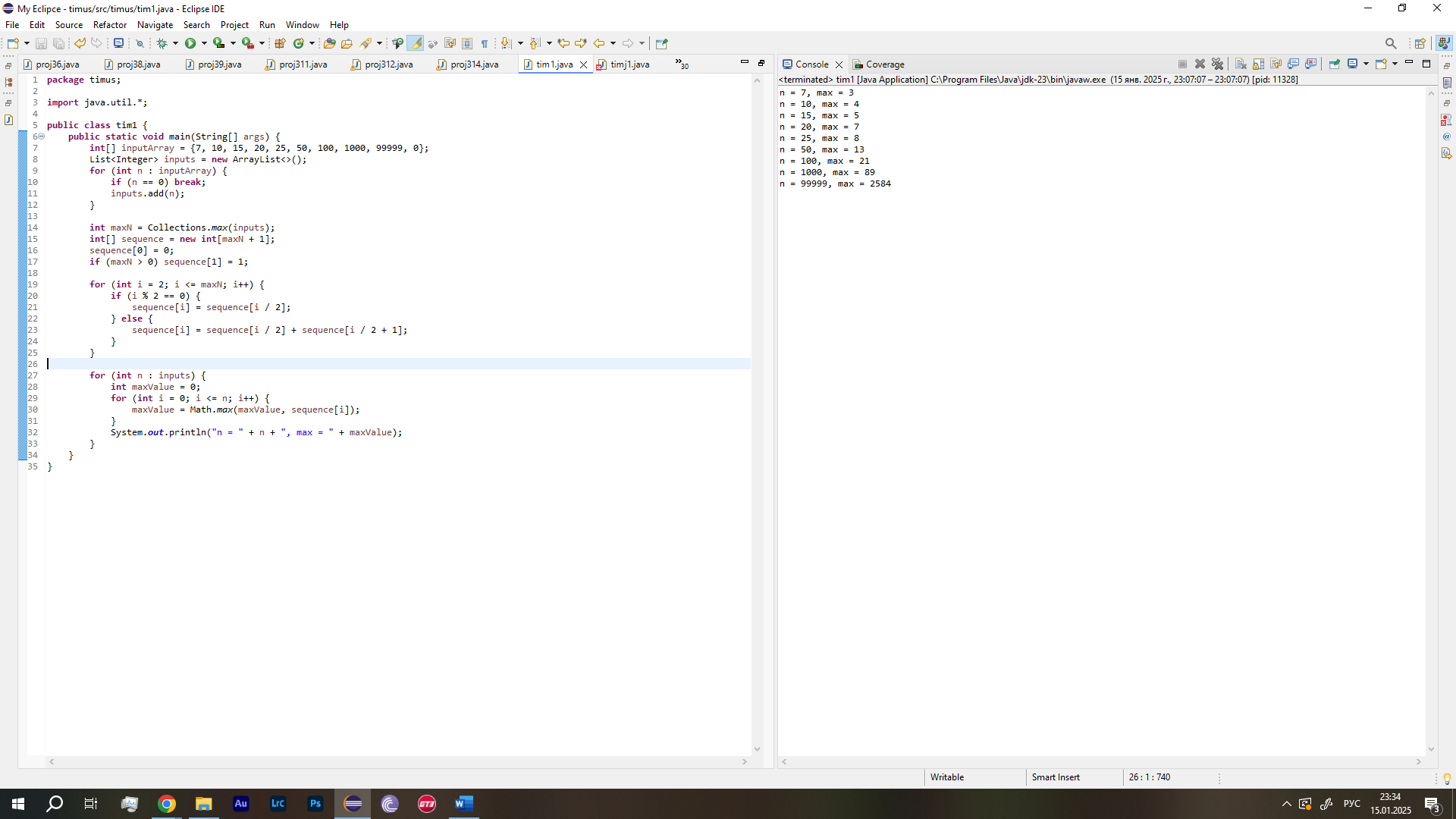


Рисунок 1 – Код и результат программы tim1

**Задание 1044**. Вводится чётное число N — количество цифр в номере билета (2 ≤ N ≤ 8). Выведите количество билетов, в номере которых сумма первой половины цифр равняется сумме второй половины цифр (количество «счастливых» билетов).

Пользователем вводится четное число от 2 до 8, иначе программа сообщает об ошибке. Далее число делится на две части. Метод countCombinations считает количество способов составить заданную сумму цифр, результат расчетов записывается в целочисленный массив dp. Результат работы программы и ее программный код представлен на рисунке 2.

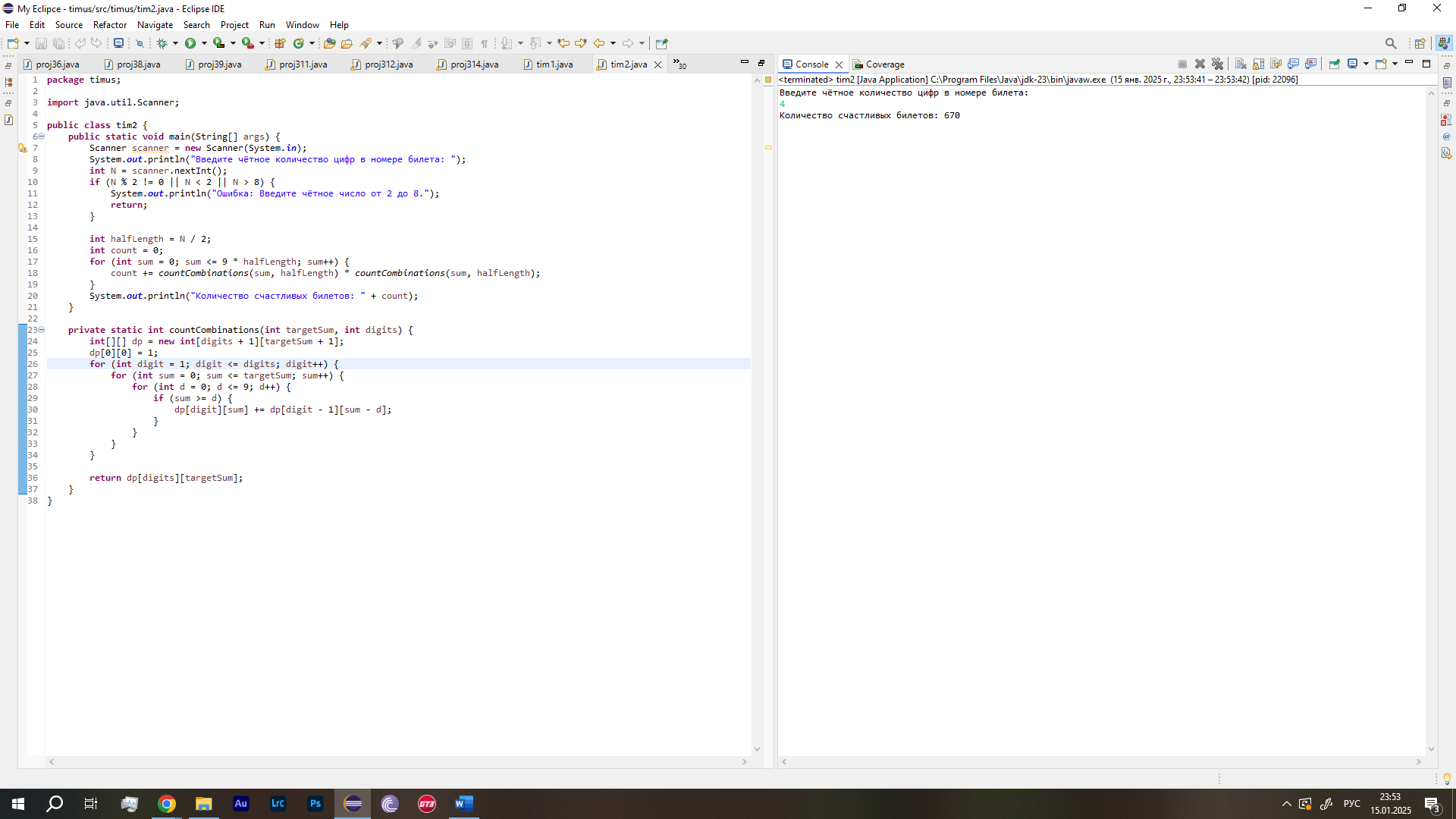


Рисунок 2 – Код и результат программы tim2

**Задание 1134**. На обеих сторонах каждой из n карт записаны числа. На первой карте 0 и 1, на второй — 1 и 2, …, на n-й — (n − 1) и n. Первоклассник Коля берёт карты по одной в случайном порядке и у каждой карты читает число на одной из её сторон. Коля не очень хорошо разбирается в числах, поэтому он мог допустить ошибку. Ваша задача — определить, ошибся ли он, то есть является ли заданная последовательность чисел возможной для некоторого порядка взятия карт.

Пользователем вводится количество карт n, длинна последовательности m, а также пользователь вводит сами числа последовательности. Каждая карта имеет на своих сторонах два числа; т.е. карта 1 – 0 и 1, карта 8 – 7 и 8, и т.д. Для каждого числа проверяется, есть ли оно в множестве чисел, если хотя бы одно число не попадает в него, выводится соответствующее сообщение. Если все числа попадают, выводится сообщение о том, что последовательность возможна. Результат работы программы и ее программный код представлен на рисунке 3.

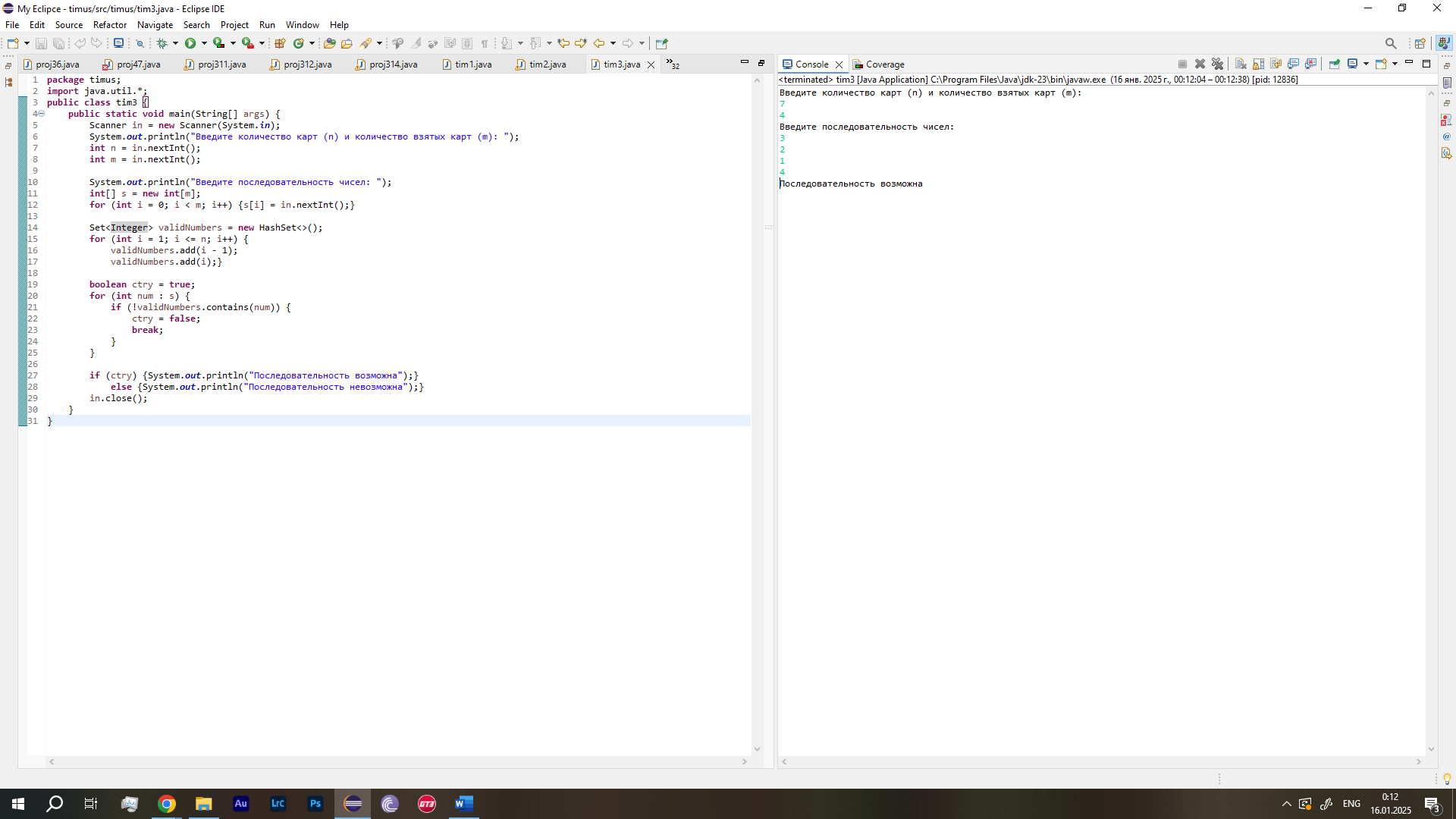


Рисунок 3 – Код и результат программы tim3

**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 4.

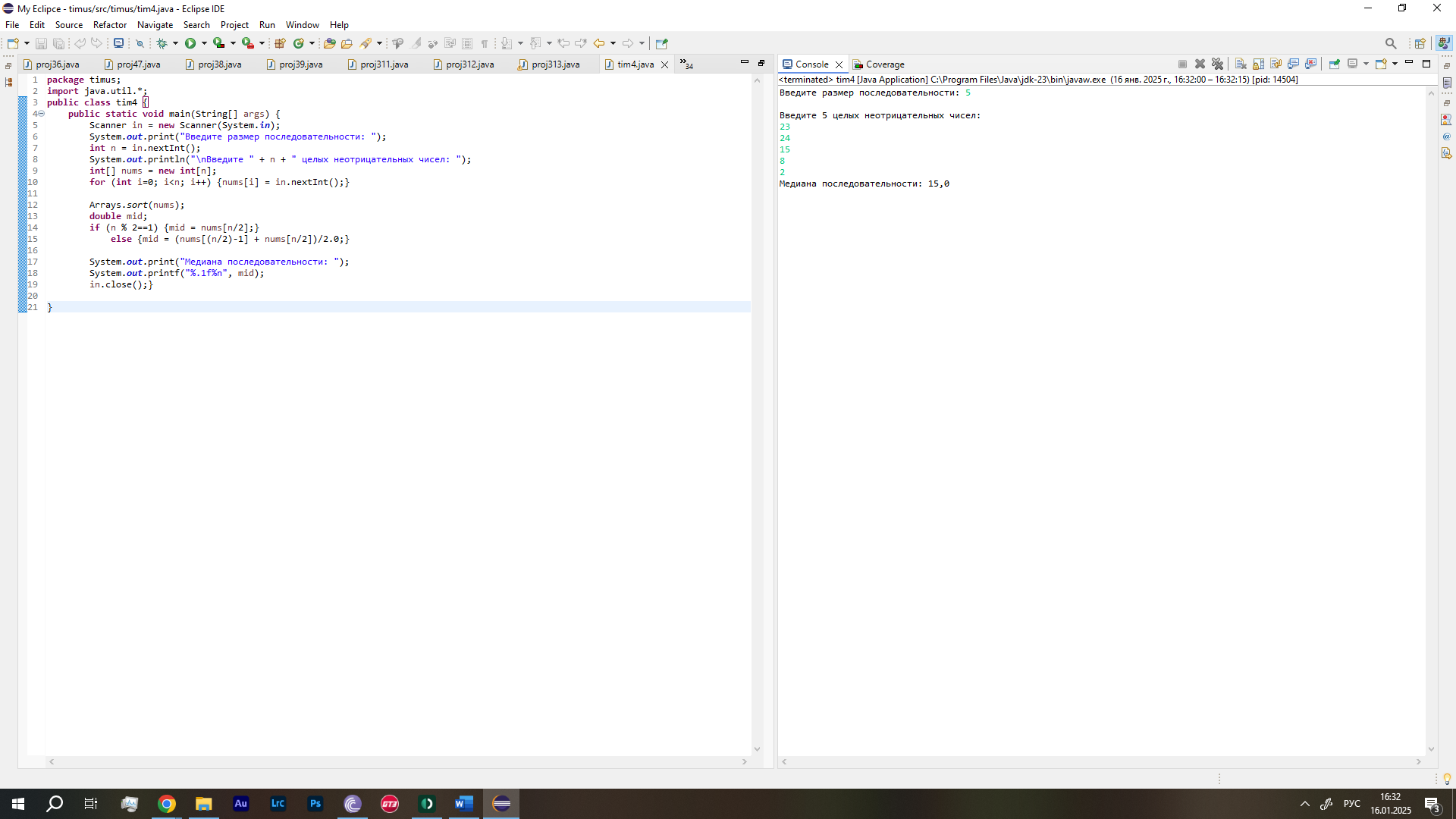


Рисунок 4 – Код и результат работы программы tim4

**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 true, если ода из введенных оценок равна 3, значение allfive равна false, если одна из оценок не равна 5. Переменная sum хранит сумму всех оценок для вычисления среднего балла. Результат выводится в консоль

Результат работы программы и ее код показан на рисунке 5.

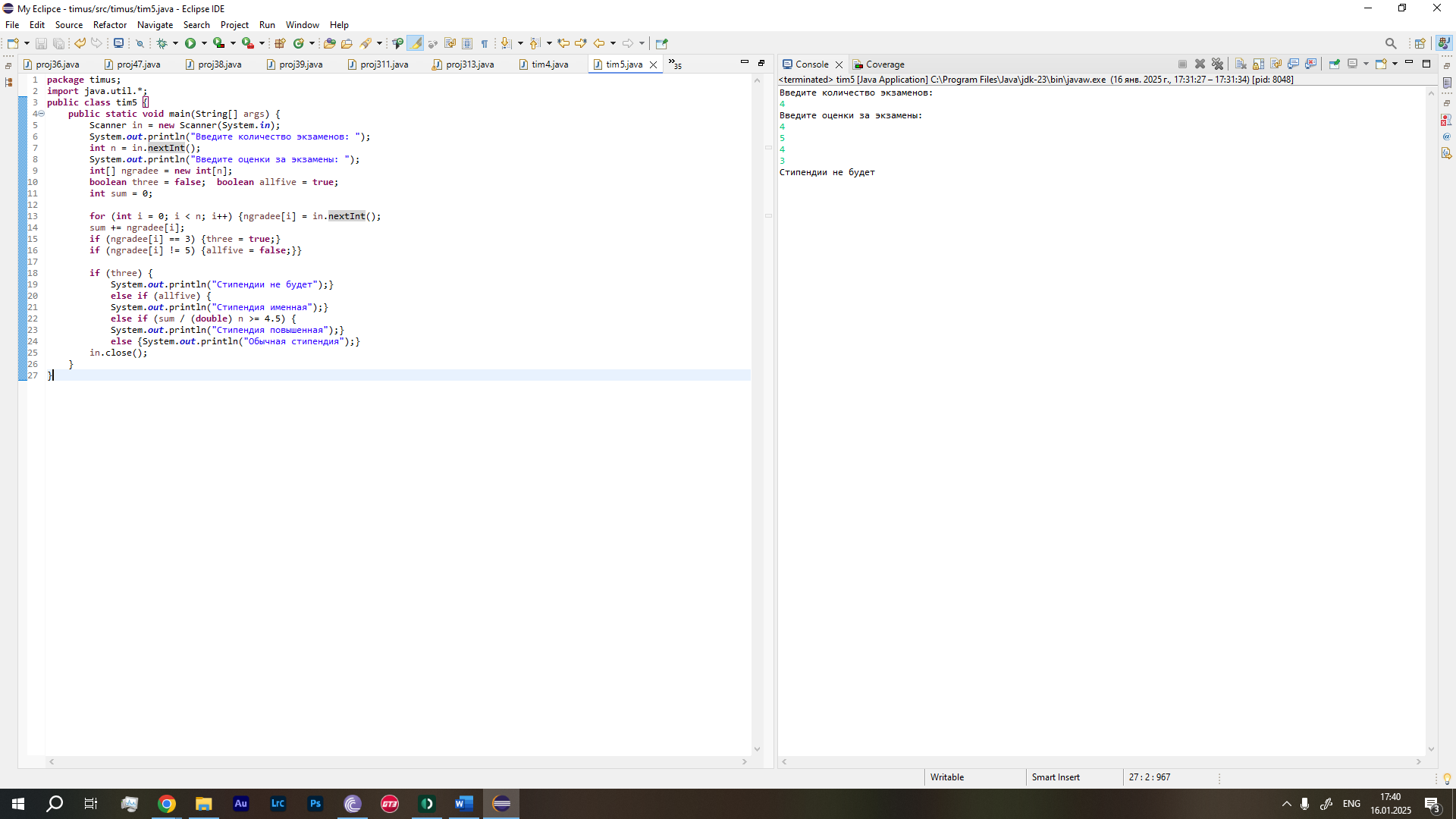


Рисунок 5 – Код и результат работы программы tim5

**Задание 1582**. Вам будут даны самые выгодные коэффициенты k1, k2 и k3 на три возможных исхода матча. Найдите максимальную сумму, которую можно гарантированно выиграть, если сделать ставки на общую сумму 1000 фунтов стерлингов.

Пользователем в консоль вводятся коэффициенты для трех возможных исходов, тип их значение – double. Сумма задается фиксированная – 1000 фунтов стерлингов. Далее высчитывается сумма, которую надо поставить для каждого исхода, чтобы получить одинаковый выигрыш при любом результате. Высчитывается гарантированный выигрыш. Результаты расчетов выводятся в консоль. Результат работы программы и ее программный код представлен на рисунке 6.

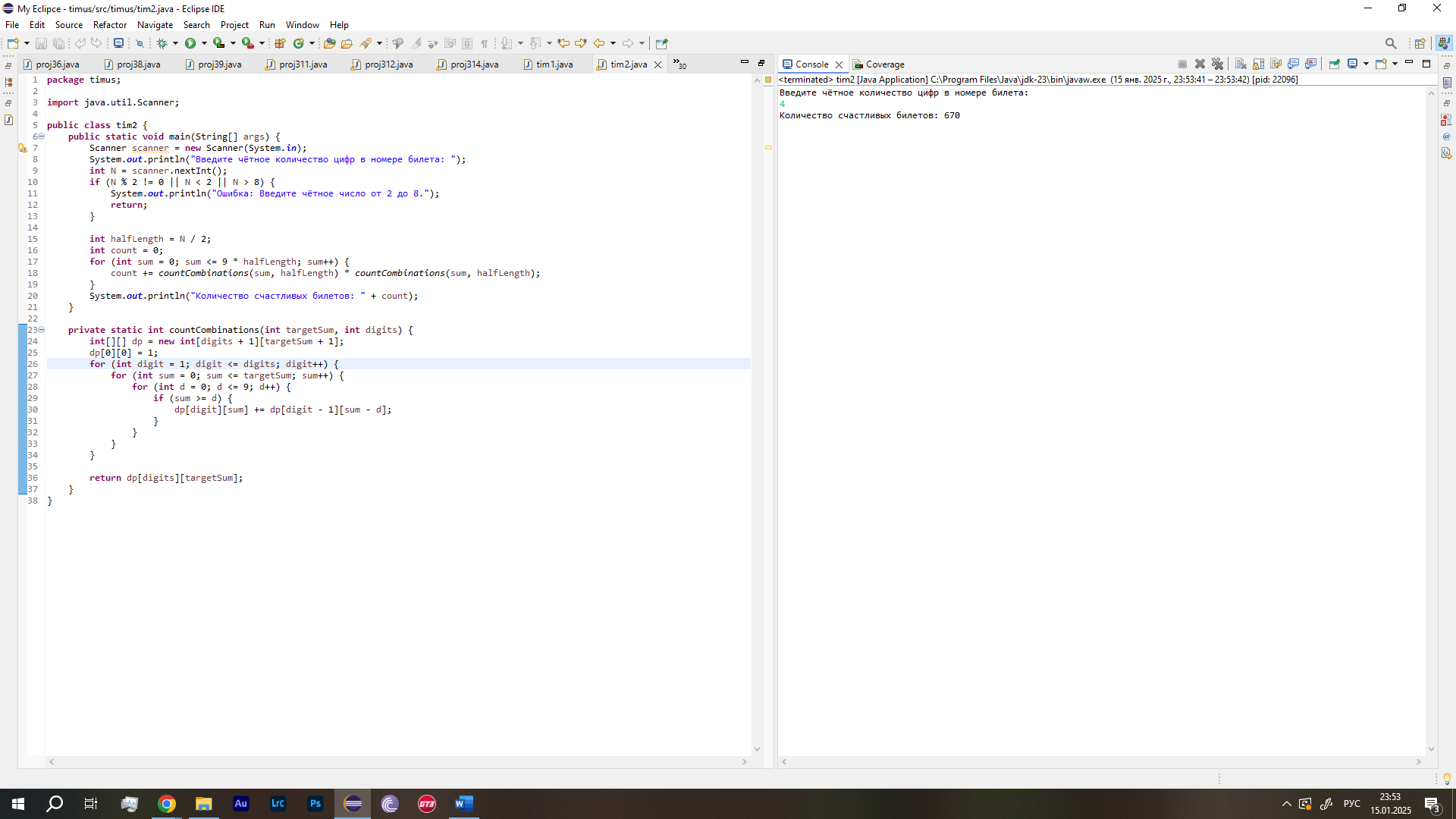


Рисунок 6 – Код и результат программы tim6

**Задание 1549**. Один из преподавателей УрГУ приобрел в Японии забавную игрушку: маленький самолетик и много пластмассовых пластинок, из которых, как из кусочков пазла, можно сложить путь для самолетика. Складывать пластинки можно как угодно, но если сложить пазл правильно, то получится карта Японии, и самолетик будет объезжать по замкнутому маршруту все крупные достопримечательности Японии. Кусочки пазла можно считать квадратами одинакового размера, путь всегда соединяет середины двух сторон квадрата. То есть кусочки бывают двух типов: прямые и повороты. В строке записаны два целых числа — количество прямых S и количество поворотов T (0 ≤ S, T ≤ 1000, S + T > 0).

Пользователем вводятся значения для двух целочисленных переменных – количество прямых участков S и количество поворотов T. Если сумма прямых поворотов меньше 4 или количество поворотов меньше 2, выводится сообщение об ошибке (Atawazu). За количество прямых поворотов отвечает uF, за количество левых и правых – uL и uR. Построение пути идет в цикле, пока есть возможность добавить прямые участки и общая сумма поворотов не превышает Т. Результат работы программы и ее программный код представлен на рисунке 7.

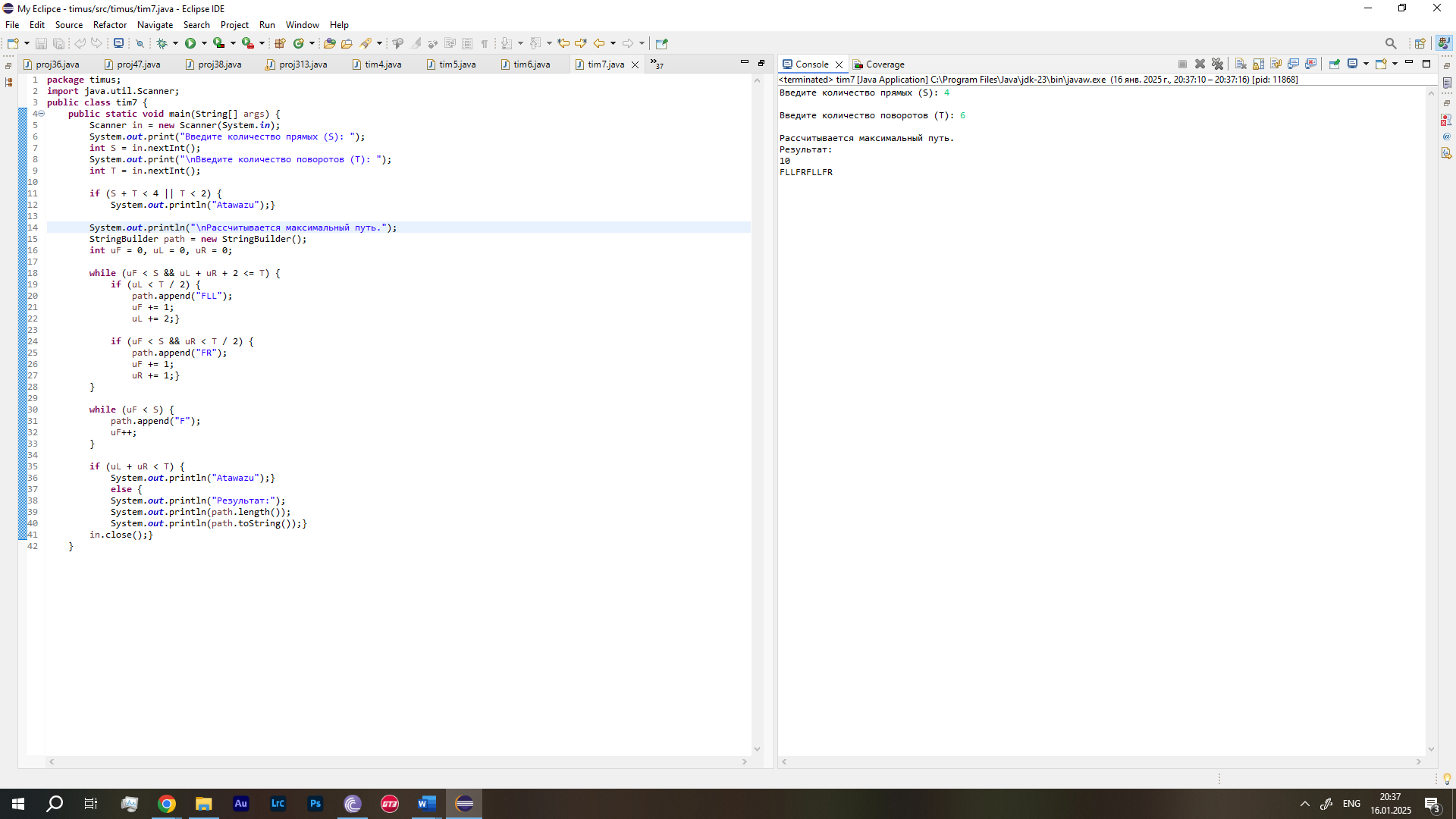


Рисунок 7 – Код и результат программы tim7

**Задание 1635.** К пересдаче Васечкин решил выучить название предмета. Чтобы лучше запомнить эту длинную строку, он решил разбить её на палиндромы и запомнить каждый из них по отдельности. Конечно, количество палиндромов в разбиении должно быть минимально возможным. В первой строке записано название предмета, который сдавал Васечкин, — непустая строка, содержащая только маленькие латинские буквы. Длина строки не превосходит 4000 символов. В первой строке выведите минимальное количество палиндромов, на которое можно разбить название предмета. Во второй строке через пробел выведите палиндромы из оптимального разбиения. Если возможных ответов несколько, выведите любой.

Пользователем вводится в консоль преобразуемая строка s, создается двумерный массив palindrome, который будет хранить значения true или false, в зависимости будет ли являться подстрока палиндромом. Если строка состоит из одного символа – это палиндром. Если длинна подстроки равна 2(i-j=1), она тоже палиндром, если символы одинаковы. Массив dp хранит минимальное количество палиндромов, а массив sindex – индекс, где заканчивается последний палиндром в разбиении s[0…i]. Затем используя sindex восстанавливаем разбиение Получившийся результат выводится в консоль. Результат работы программы и ее код показан на рисунке 8.

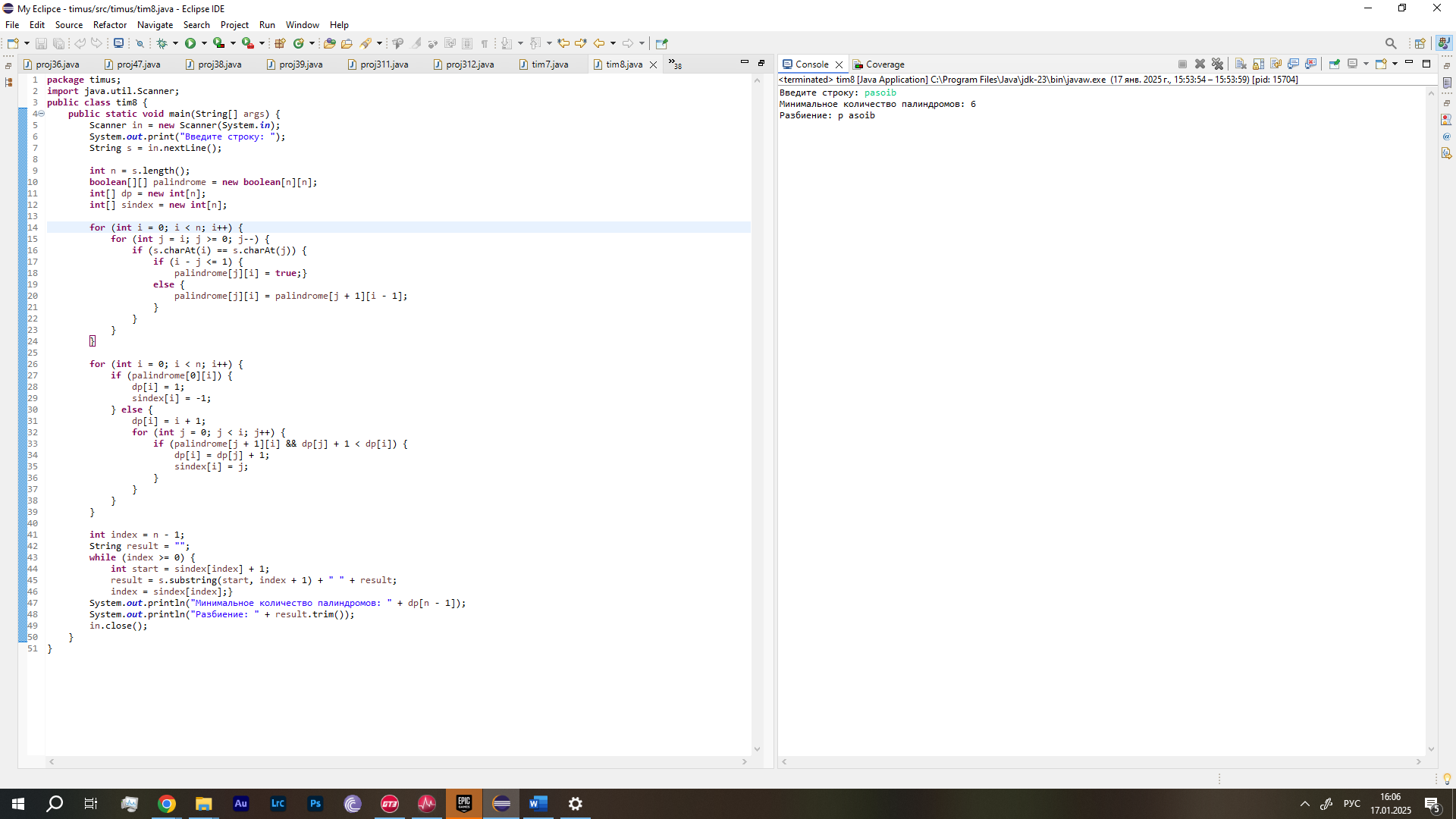


Рисунок 8 – Код и результат работы программы tim8

**Задание 1601**. У блондинки Анжелы новое увлечение — интернет-чаты. Она, как любая уважающая себя блондинка, привыкла писать свои сообщения в верхнем регистре. Входные данные содержат сообщение Анжелы, которое состоит из латинских букв в верхнем регистре, пробелов, переводов строк и знаков препинания: точек, запятых, тире, двоеточий, восклицательных и вопросительных знаков. Общая длина сообщения не превосходит 10000 символов. Выведите исправленное сообщение Анжелы.

Пользователем вводится строка символов string input, создается строка для вывода результата и «флаг», означающий что начинается новое предложение. Затем программой проверяется каждый символ: если он не стоит в начале нового предложения, символ заменяется на аналогичный с нижним регистром (функция взаимодействия с ASCII символами). Если программой встречается знак конца предложения, значение счетчика инкрементируется, а следующее предложение будет с заглавной буквы. Результат работы программы и ее программный код представлен на рисунке 9.

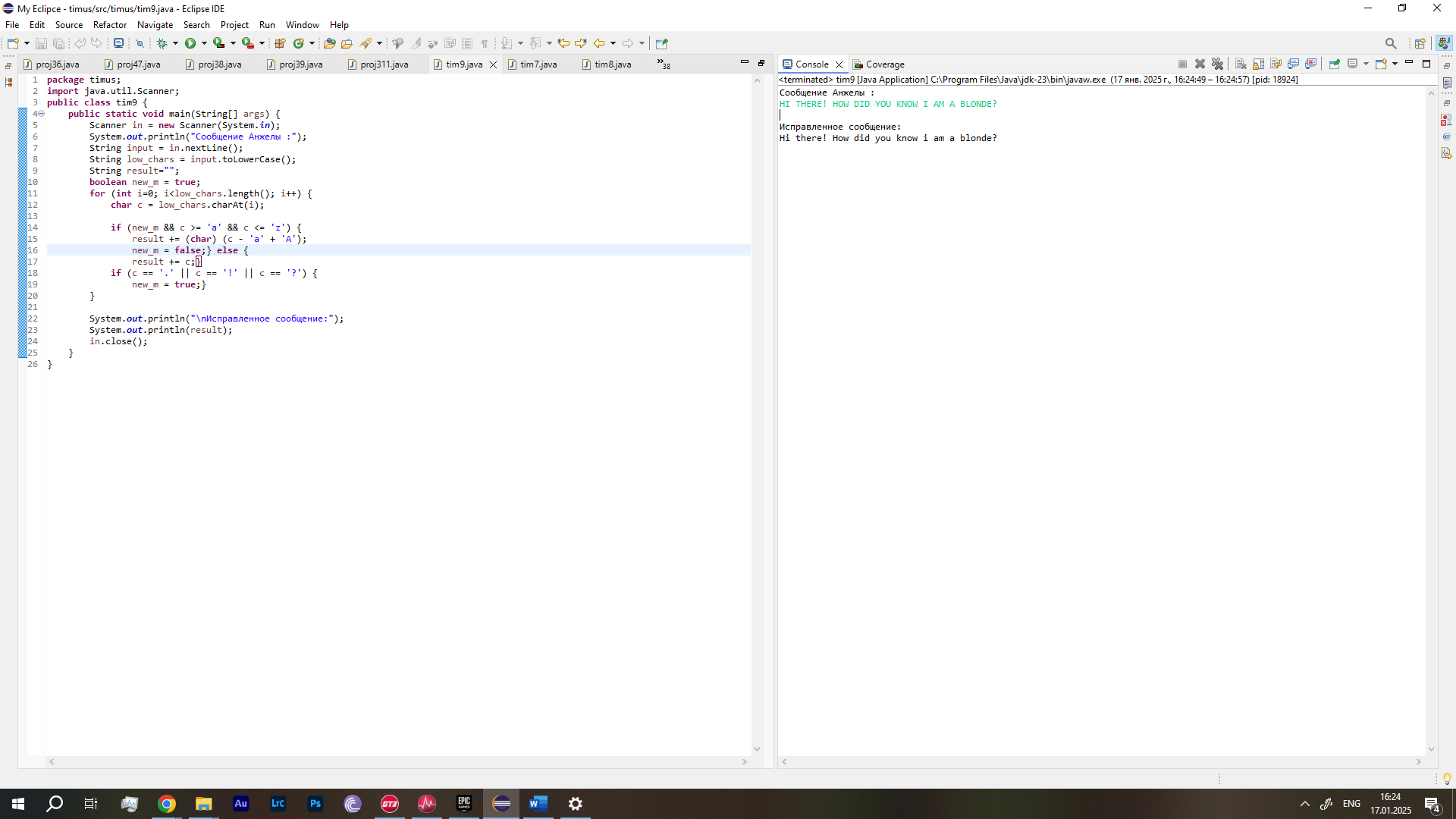


Рисунок 9 – Код и результат программы tim9

**Задание 1180**. Вам будут даны самые выгодные коэффициенты k1, k2 и k3 на три возможных исхода матча. Найдите максимальную сумму, которую можно гарантированно выиграть, если сделать ставки на общую сумму 1000 фунтов стерлингов.

Два Никифора играют в следующую игру. Перед ними лежит кучка из N камней. Никифоры по очереди берут из неё некоторое число камней. За один ход разрешается взять любое число камней, являющееся целой неотрицательной степенью числа 2 (то есть, 1, 2, 4, 8, и т.д.). Выигрывает Никифор, взявший последний камень. Требуется написать программу, которая определяла бы, какой Никифор выигрывает при правильной игре: начинающий или его партнер. В единственной строке находится целое положительное число N. В случае если выигрывает начинающий Никифор, во второй строке должно быть указано минимальное число камней, которое он должен взять первым ходом, чтобы гарантировать себе выигрыш.

Создается целочисленная переменная N – количество камней, массив используется для хранений значений true и false (для каждого шага определяет, в выигрышной или нет позиции остается текущий игрок). При победе игрока 1 или 2 в консоль выводится соответствующее сообщение, также выводится минимальное количество камней для победы. Результат работы программы и ее программный код представлен на рисунке 10.

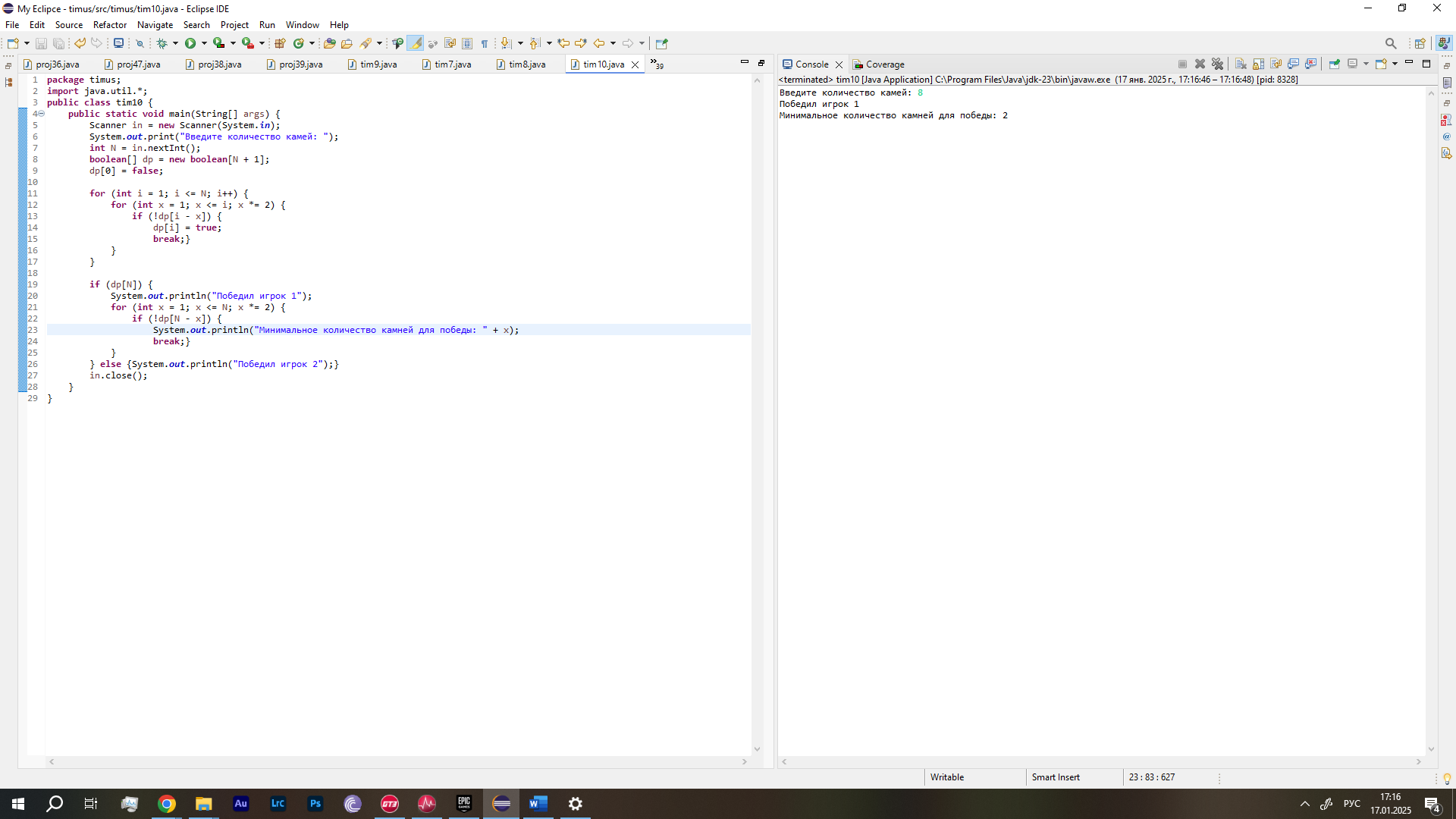


Рисунок 10 – Код и результат программы tim10

**Задание 1713**. Рассмотрим какую-либо команду консольной утилиты. Подстроку этой команды, которая не является подстрокой никакой другой команды утилиты, можно назвать ключевой, поскольку она однозначно идентифицирует команду. В последних версиях утилиты можно не вводить команду полностью, достаточно лишь ввести любую ключевую подстроку этой команды. В первой строке записано целое число n (2 ≤ n ≤ 1000) — количество команд консольной утилиты. В следующих n строках записаны эти команды — непустые строки, состоящие из строчных латинских букв. Никакая команда не является подстрокой другой команды.

Выведите n строк. В i-й строке должна быть записана любая из самых коротких ключевых подстрок i-й команды (команды занумерованы в том порядке, в котором они даны на входе)..

Программа находит минимальную ключевую подстроку. В первой строке вводится количество команд n, далее вводятся сами команды в количестве n (могут быть только из латинских символов). Для каждой команды находится минимальная подстрока, будущая уникальной, сами команды хранятся в массиве commands. В функции find\_nim\_key осуществляется поиск минимальной ключевой строки. Big\_l используют для перебора длин подстрок команды, start указывает на начальную позицию для извлечения подстроки из command, а в sub хранится текущая подстрока, извлеченная из команды. Переменная unique в свою очередь проверяет уникальность этих (извлеченных) команд. Ввод и вывод сопровождается сообщениями для пользователя, выводимыми в консоль. Результат работы программы и ее программный код представлен на рисунке 11.

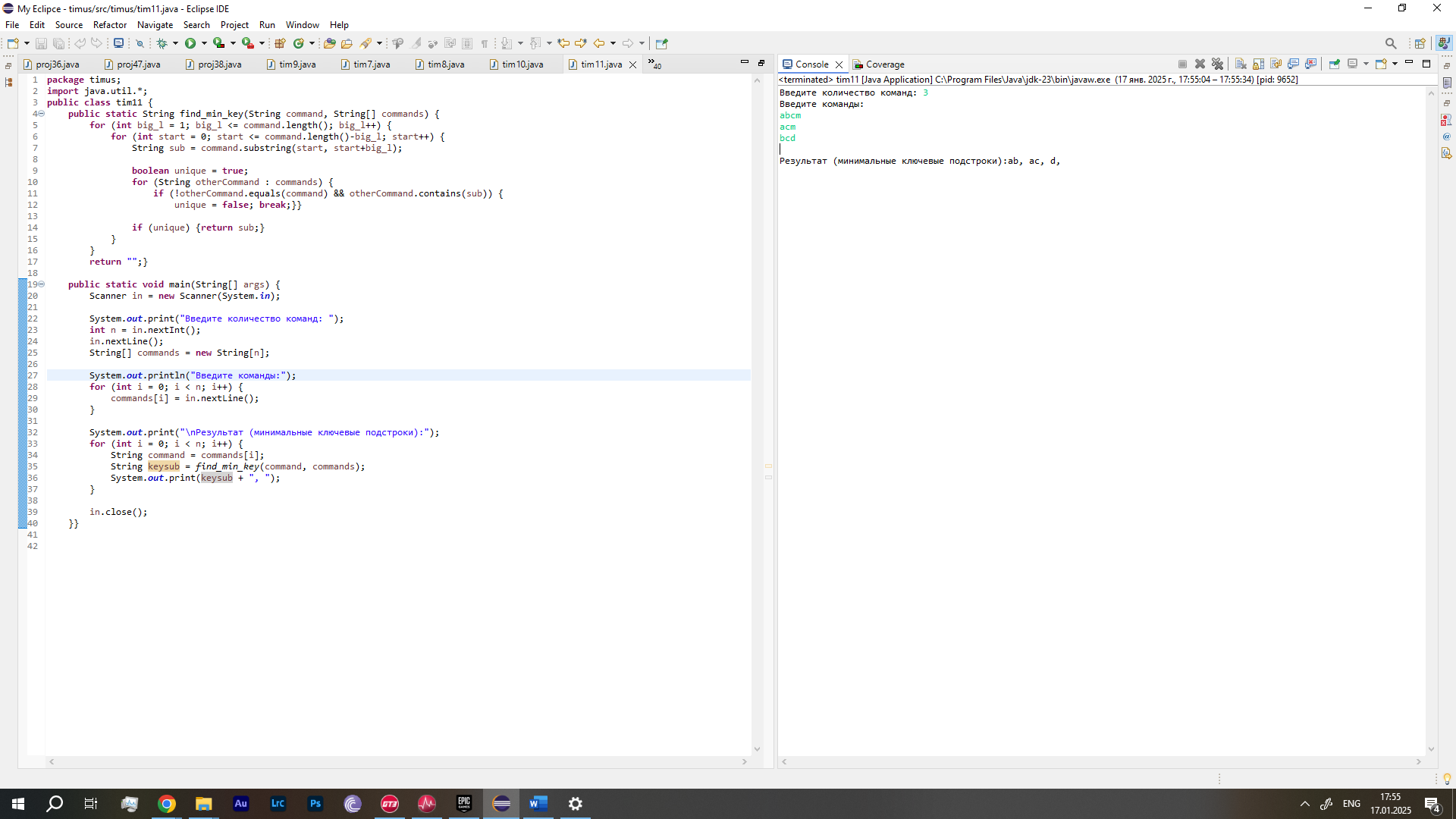


Рисунок 11 – Код и результат программы tim11