Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа «Компьютерных технологий и информационных систем»

**ОТЧЕТ по лабораторной работе № 9**

по дисциплине «Системный подход в разработке программного обеспечения»

**Выполнил:**

студент группы 5130902/20201 \_\_\_\_\_\_\_\_\_\_\_\_\_ А. И. Сафонов

подпись

**Проверил:**

кандидат тех. наук, доцент \_\_\_\_\_\_\_\_\_\_\_\_ С. А. Нестеров

подпись

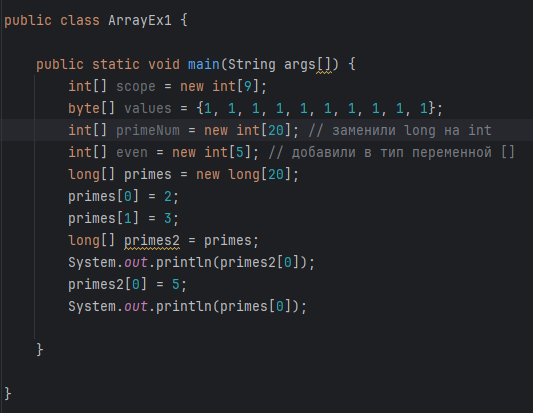
«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024г.

Санкт-Петербург

2024

1. **По занятию 8.1**
2. **Упражнение 2**

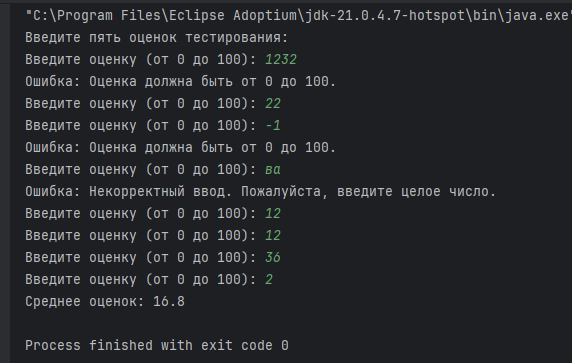
* Импортируйте и откройте проект ArrayEx.
* Проверьте ArrayEx1.java.
* Измените программу, чтобы реализовать следующее…
* Объявить одномерный массив с именем score типа int, который содержит 9 значений.
* Объявить и инициализировать одномерный массив байтов с именем values и размером 10, чтобы все записи содержали 1.
* Удалить комментарии для двух строк, которые были прокомментированы, затем устранить ошибки синтаксиса.

****

1. **Упражнение 3**

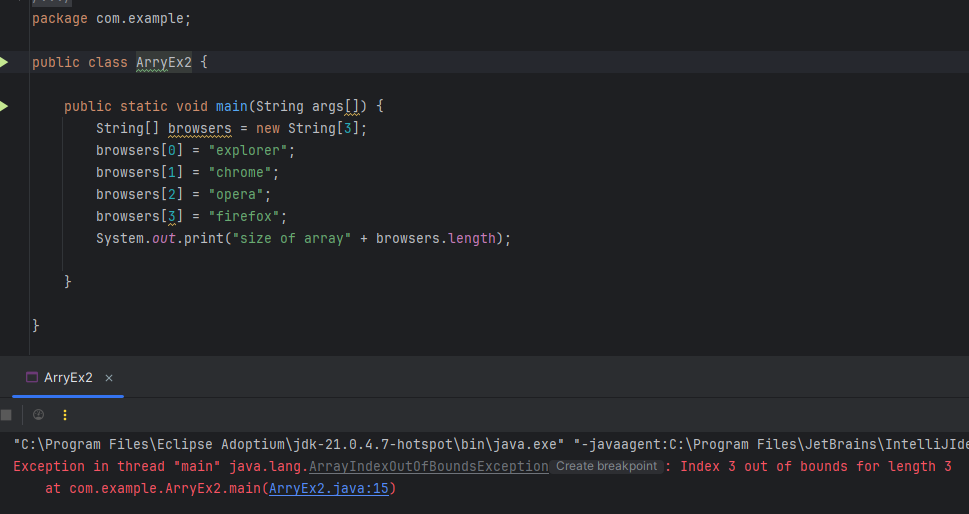
* Импортируйте и откройте проект ArrayEx.
* Проверьте ComputeAvg.java.
* Измените программу, чтобы реализовать следующее…
* В определенном классе представлено пять тестов, каждый из которых оценивается в 100 баллов.
* Введите пять оценок тестирования из консоли.
* Сохраните оценки тестирования в массиве.
* Вычислите средние оценки успеваемости студентов.

****

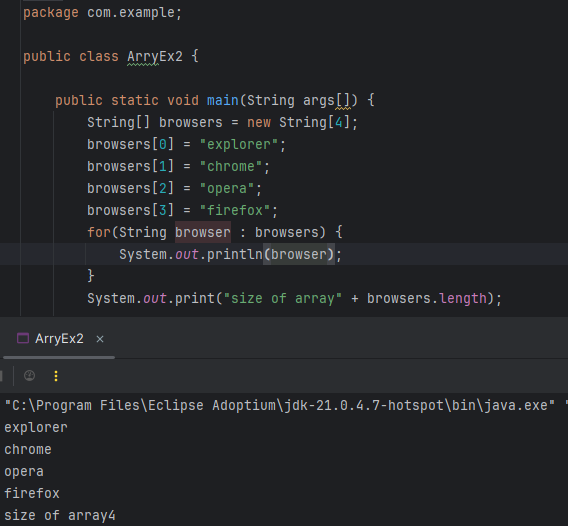
****

1. **Упражнение 4**

* Импортируйте и откройте проект ArrayEx.
* Проверьте ArrayEx2.java.
* Выполните следующие действия.
* Запустите программу и посмотрите на ошибку.
* Измените программу для устранения ошибки.
* С помощью цикла for-each отобразите все браузеры, которые сохранены в массиве.

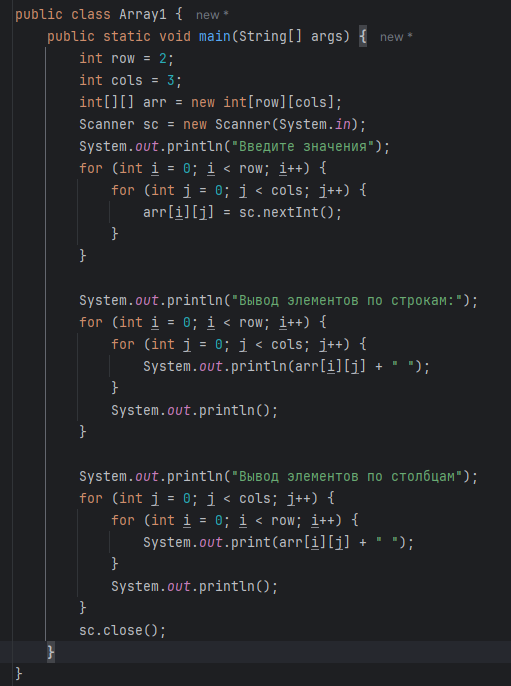


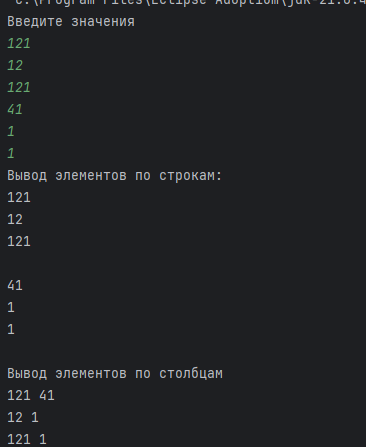
Возникает исключение ArrayIndexOutOfBoundsException, так как массив имеет индексы от 0 до 2 и пытается получить доступ к элементу с индексом 3.



1. **Задание 1**

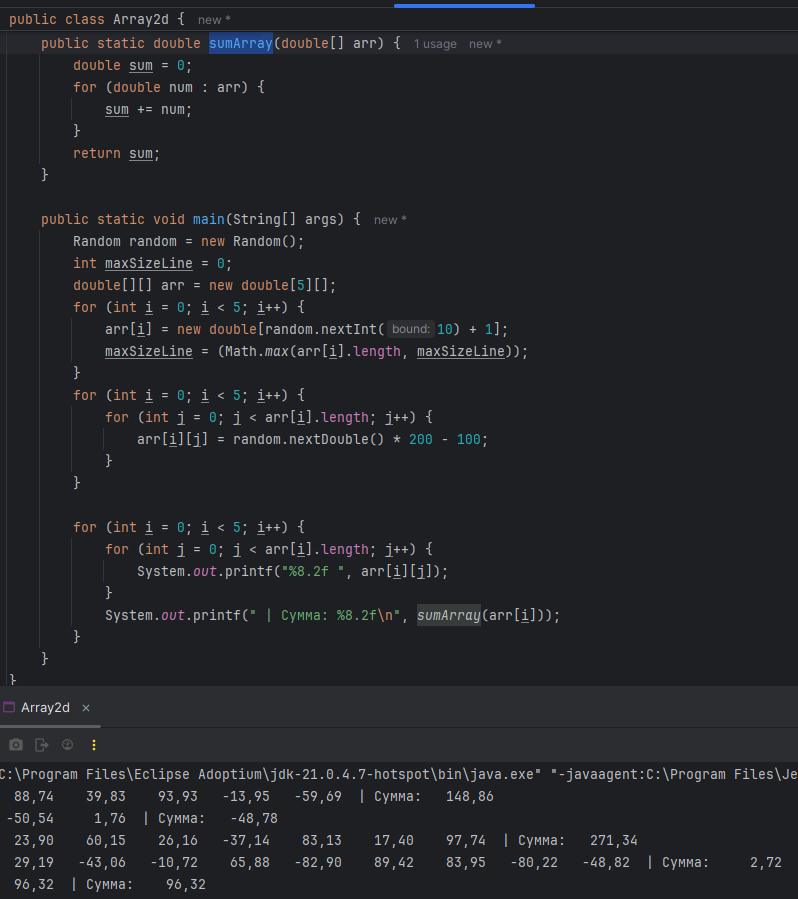
Объявите двумерный массив с элементами типа int, двумя строками и тремя столбцами. Заполните его значениями, введенными с клавиатуры. Сначала выведите элементы массива по строкам (т.е. в консоли напечатайте 2 строки, 3 столбца), потом выведите его по столбцам (т.е. 3 строки и 2 столбца).





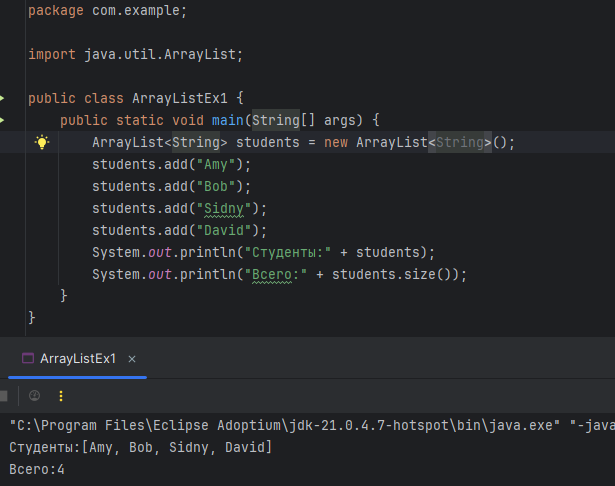
1. **Задание 3**

Объявите двумерный массив с элементами типа double и 5 строками. Число элементов в каждой строке выбираете случайным образом из диапазона от 1 до 10 (см. книгу Васильева про двумерные массивы со строками разной длины). Заполните массив случайными вещественными числами из диапазона от -100 до 100. Напишите в главном классе статический метод, считающий сумму элементов одномерного массива (полученного в качестве параметра). Выведите в консоль заполненный массив по строкам и после каждой строки выведите сумму ее элементов, рассчитанную вашим методом.

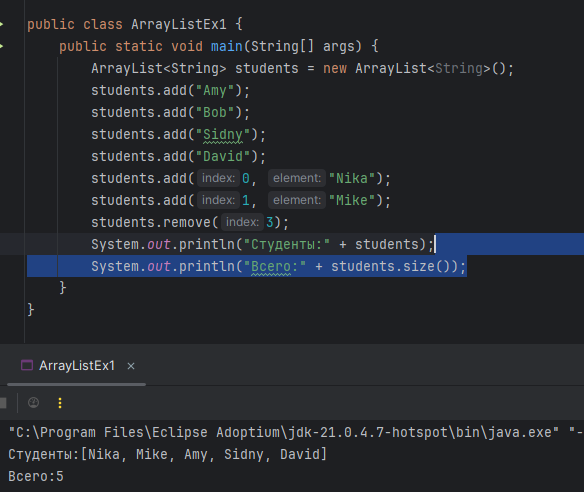


1. **По занятию 8.2**
2. **Упражнение 1**

* Импортируйте и откройте проект ArrayListsEx.
* Проверьте ArrayListEx1.java.
* Измените программу, чтобы реализовать следующее:
* Создайте массив ArrayList для String с названием "Студенты".
* Добавьте четырех студентов в ArrayList: Эйми, Боб, Синди и Дэвид.
* Вывод элементов ArrayList и отображение размера списка.

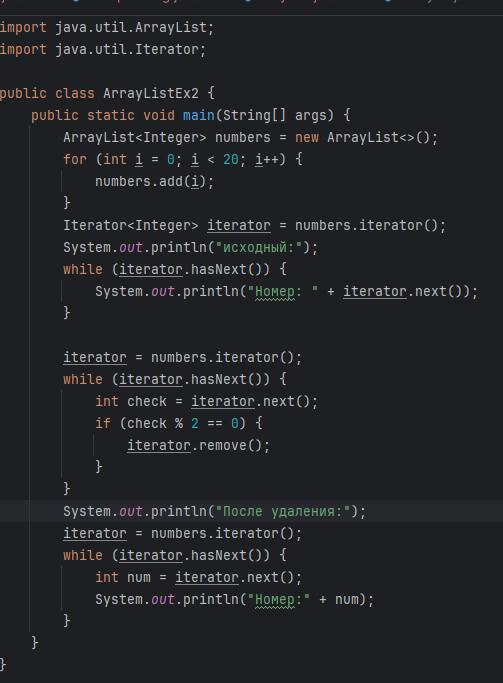


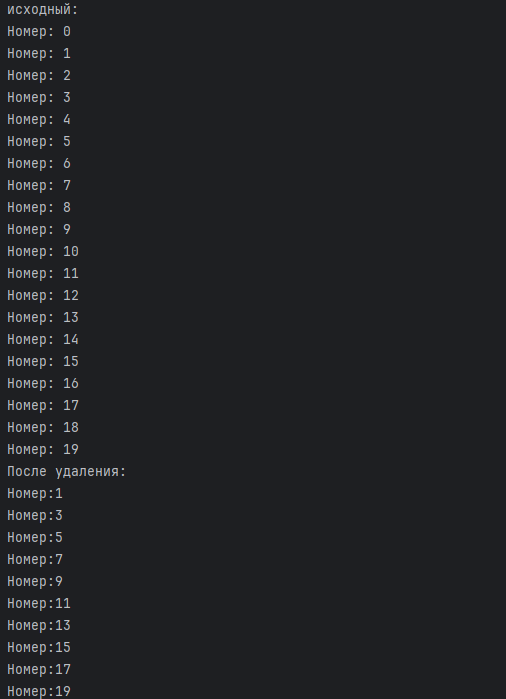
* Измените программу, чтобы реализовать следующее:
* Добавьте двух других студентов (Ника и Майка) в индекс 0 и 1.
* Удалите студента в индексе 3.
* Вывод элементов ArrayList и отображение размера списка.



1. **Упражнение 2**

* Импортируйте и откройте проект ArrayListsEx.
* Проверьте ArrayListEx2.java.
* Выполните следующие действия.
* Создайте ArrayList со списком номеров.
* Отобразите содержимое ArrayList с помощью Iterator.
* Удалите все четные числа.
* Отобразите содержимое ArrayList.





1. **По уроку 8.3**
2. **Упражнение 1**

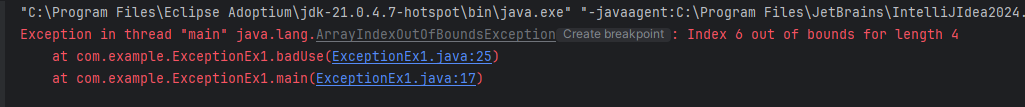
• Импортируйте и откройте проект ExceptionsEx.

• Проверьте ExceptionEx1.java:

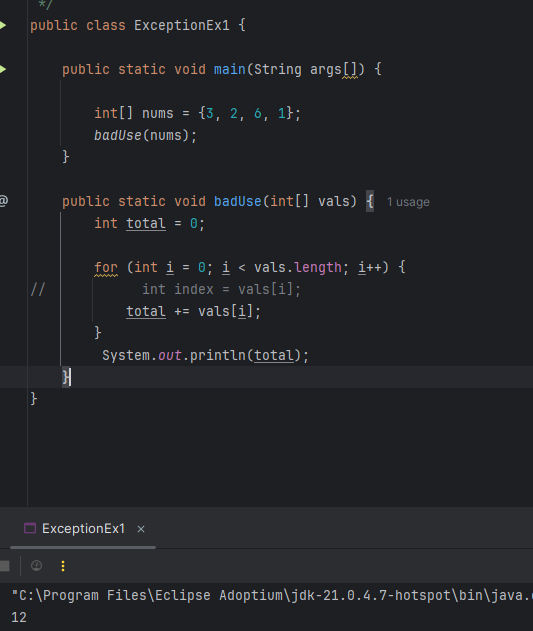
– Выполните программу и посмотрите на результат:

– Возникает исключение ArrayIndexOutOfBoundsException.

– Рекомендуется ли обрабатывать исключения для данной программы?

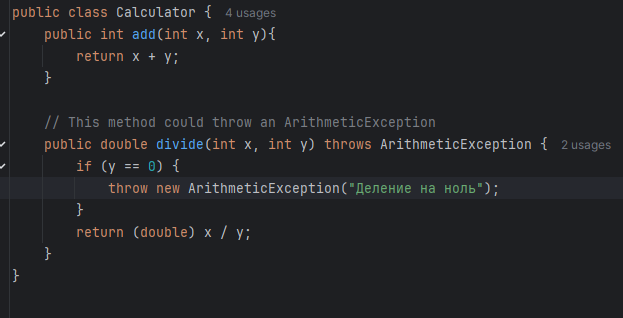
– Измените программу для вычисления суммы массива.

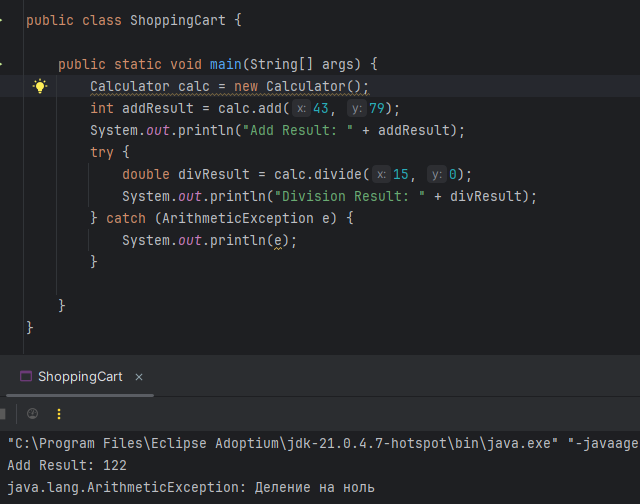
Ошибка заключается в том, что используются значения из массива vals как индексы для доступа к элементам. Это может привести к попытке получения доступа к несуществующему индексу массива (ArrayIndexOutOfBoundsException) если значение в vals больше, чем максимальный допустимый индекс.



1. **Упражнение 2**

* Импортируйте и откройте проект ExceptionsEx.
* Проверьте Calculator.java и ShoppingCart.java.
* Измените программы, чтобы реализовать обработку исключения:
* Calculator.java:
* Определяет возможное исключение.
* Изменяет сигнатуру метода разделения для указания, что он выдает исключение.
* ShoppingCart.java:
* Захватывает исключение в классе, который вызывает метод разделения.

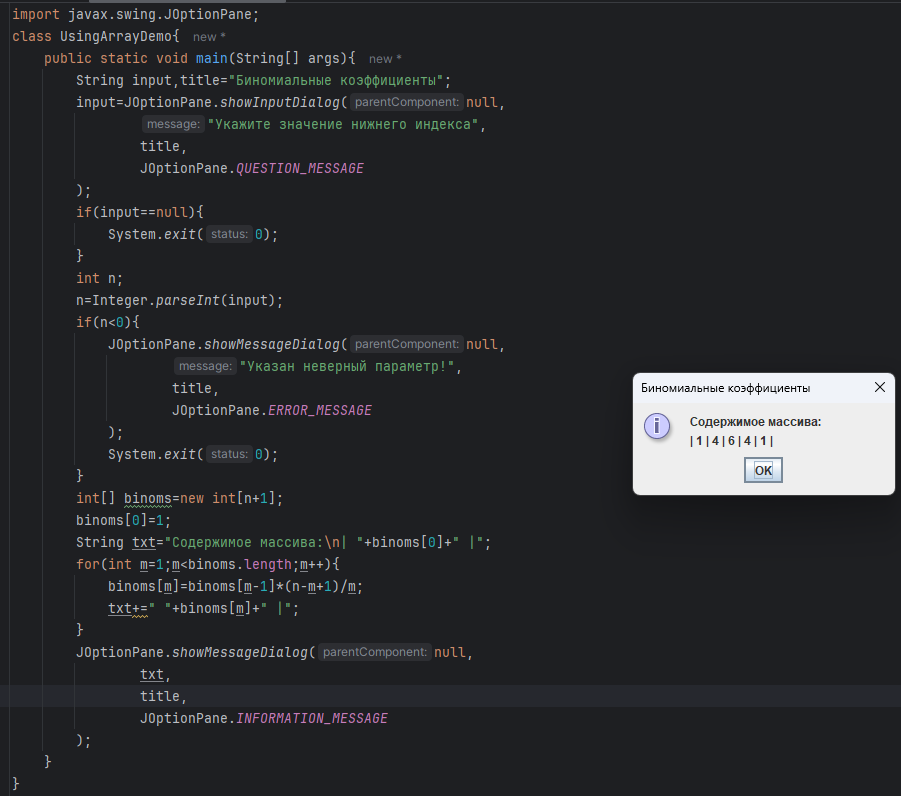
****



1. **Примеры из книги**[**Васильев, А. Н. Программирование на Java для начинающих**](https://dl.spbstu.ru/mod/resource/view.php?id=48001)

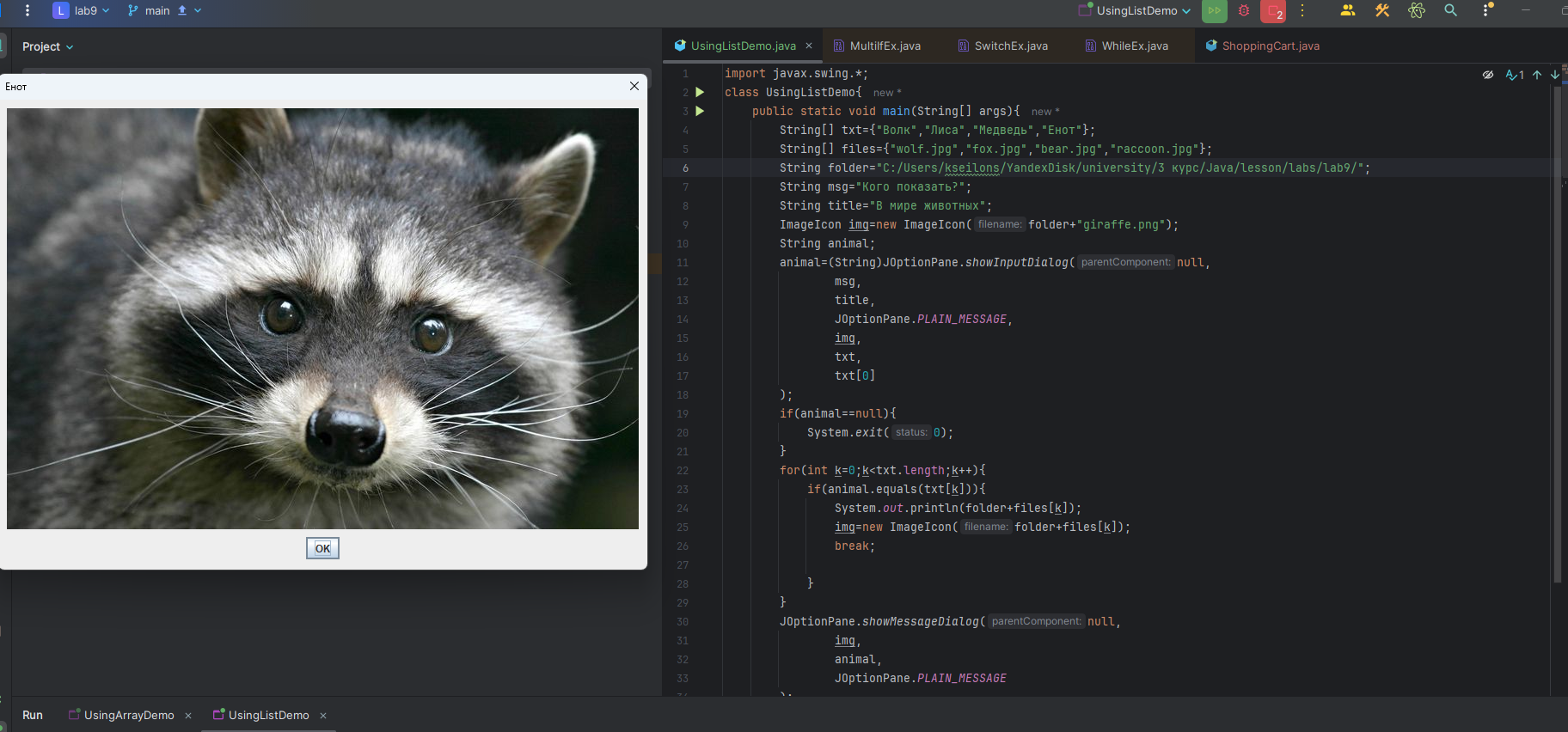
**Листинг 5.1. Программный код проекта UsingArrayApplication**

В представленной программе на основе неотрицательного целочисленного значения, которое вводит пользователь в поле ввода, создается целочисленный массив, который заполняется биномиальными коэффициентами. Затем значения элементов массива отображаются в диалоговом окне.



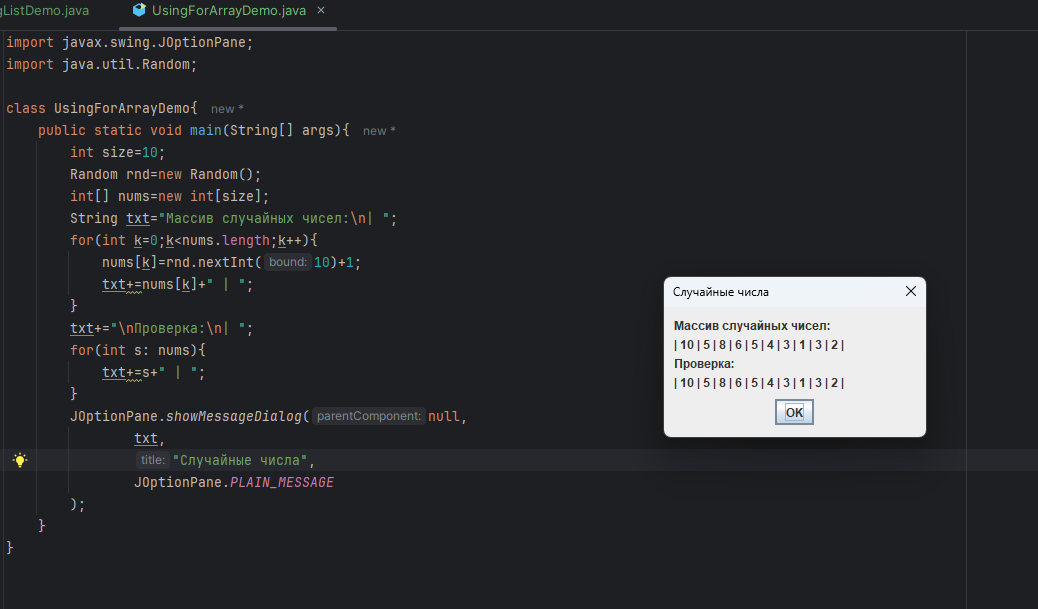
**Листинг 5.2. Программный код проекта UsingListApplication**

Как пример объявления и инициализации массивов рассмотрим пример, в котором сначала отображается диалоговое окно с раскрывающимся списком. В раскрывающемся списке выбирается название животного (волк, лиса, медведь или енот), после чего открывается новое диалоговое окно с соответствующим изображением. В программе мы реализуем названия животных и названия файлов с их изображениями в виде текстовых массивов (элементами массива являются текстовые значения). При этом массивы при объявлении инициализируются списками из текстовых значений.



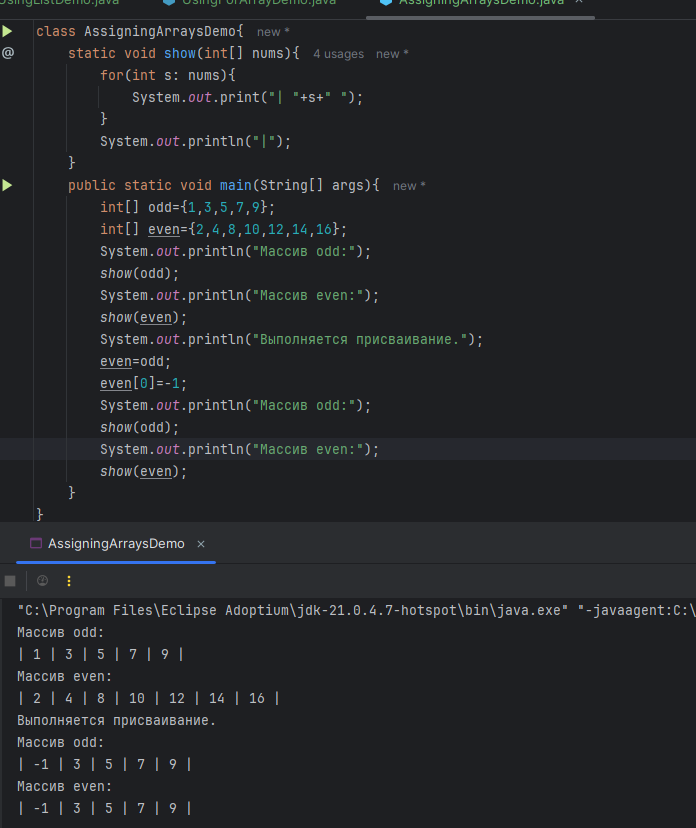
**Листинг 5.3. Программный код проекта UsingForArrayApplication**

Создается целочисленный массив, и затем он заполняется случайными числами (в диапазоне возможных значений от 1 до 10). Содержимое массива отображается в диалоговом окне. При отображении содержимого массива мы используем два разных оператора цикла, выполняя перебор как индексов элементов, так и непосредственно элементов массива.



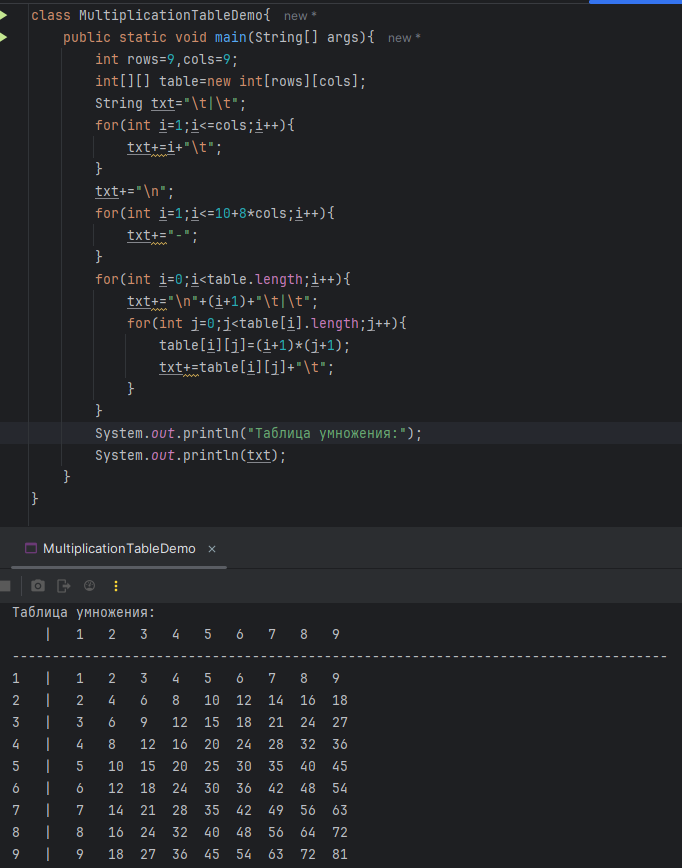
**Листинг 5.4. Программный код проекта**

В программе (в главном методе) объявляются и инициализируются два целочисленных массива odd и even. Для начала содержимое обоих массивов отображается в консольном окне. Для отображения содержимого массива использован статический метод show(), не возвращающий результат. Аргументом методу передается массив — точнее, переменная массива (команды show(odd) и show(even)). Аргумент метода, являющийся переменной массива, описывается точно так же, как объявляется переменная массива: после идентификатора типа элементов массива указываются пустые квадратные скобки. После отображения содержимого массивов odd и even командой even=odd выполняется присваивание массивов (точнее, присваивание переменных массива), а командой even[0]=-1 изменяется значение первого элемента массива. Затем снова отображается содержимое массивов odd и even. После выполнения команды even=odd и переменная even, и переменная odd ссылаются на один и тот же массив — массив, на который первоначально ссылалась переменная odd.



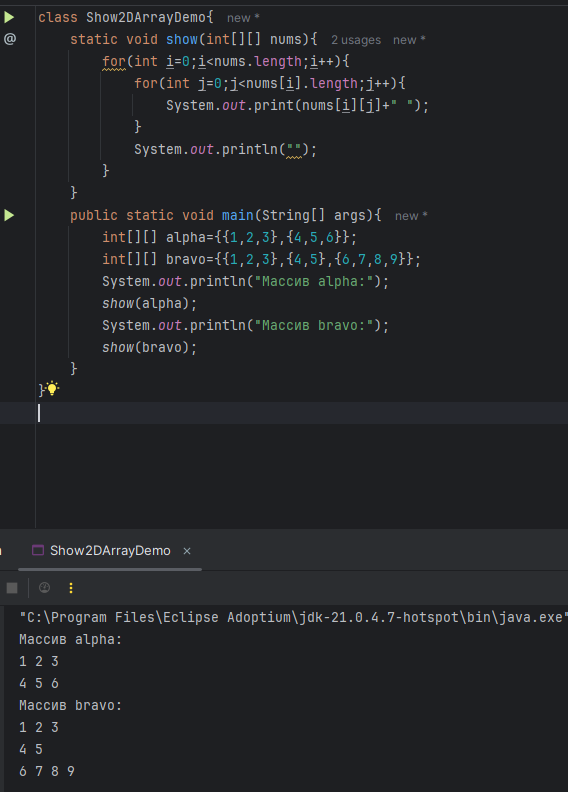
**Листинг 5.5. Программный код проекта MultiplicationTableApplication**

В данной программе в консольное окно выводится таблица умножения. В первой строке и первом столбце таблицы размещается ряд натуральных чисел от 1 до 9, а на пересечении строк и столбцов находится результат произведения соответствующих чисел в первой строке и первом столбце.



**Листинг 5.6. Программный код проекта Show2DArrayApplication**

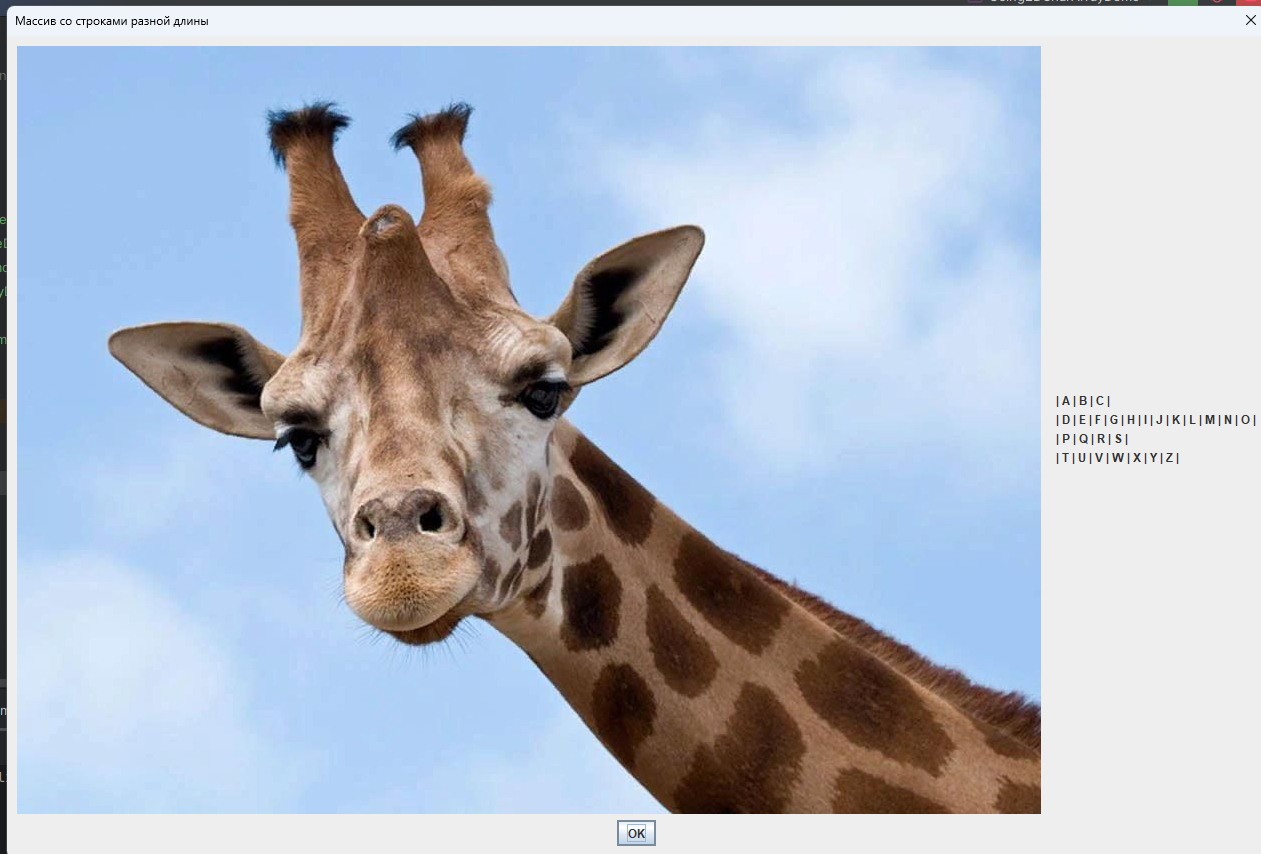
В главном методе программы командами int[][] alpha={{1,2,3},{4,5,6}} и int[][] bravo={{1,2,3},{4,5},{6,7,8,9}} объявляются и инициализируются два двумерных целочисленных массива. Командой int[][] alpha={{1,2,3},{4,5,6}} создается двумерный массив alpha из двух строк и трех столбцов. Значения элементов в первой строке массива равны 1, 2 и 3, а значения элементов во второй строке равны 4, 5 и 6. С помощью команды show(alpha) содержимое массива выводится в окно вывода. Командой int[][] bravo={{1,2,3},{4,5},{6,7,8, 9}} создается двумерный массив bravo, состоящий из трех строк. Но количество элементов в каждой строке различно. Первая строка стоит из трех элементов со значениями 1, 2 и 3. Вторая строка состоит из двух элементов со значениями 4 и 5. Третья строка состоит из четырех элементов со значениями 6, 7, 8 и 9. Содержимое массива отображается с помощью команды show(bravo).



**Листинг 5.7. Программный код проекта Using2DCharArrayApplication**

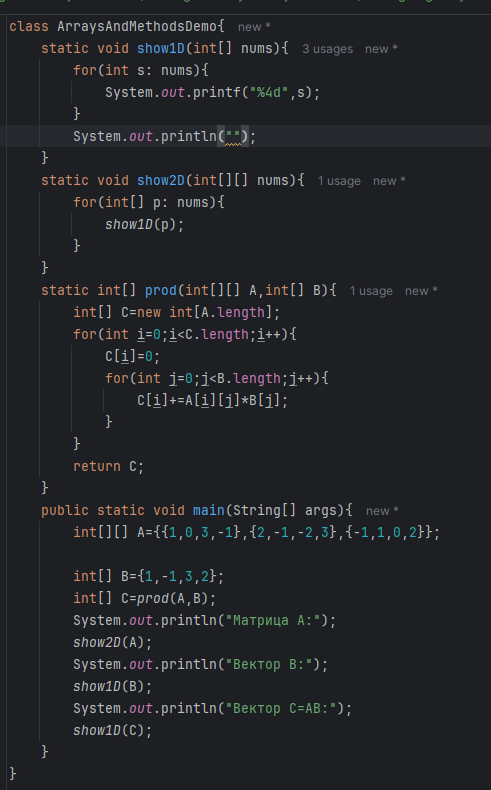
Выше мы создали двумерный массив со строками разной длины. При этом мы инициализировали массив при создании. Но процесс создания такого массива можно реализовать и несколько иначе. Как пример в листинге 5.7 приведена программа, в которой создается двумерный символьный массив со строками разной длины. Массив заполняется символами в алфавитном порядке. Количество элементов в разных строках определяется значениями элементов в целочисленном массиве.

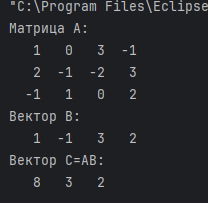




**Листинг 5.8. Программный код проекта ArraysAndMethodsApplication**

Рассмотрим математическую задачу о вычислении произведения матрицы на вектор, результатом которого является другой вектор. Векторы реализуются в виде одномерных числовых массивов, а матрица реализуется в виде двумерного числового массива.

****

****

1. **Следы чтения книги «**[**А.А. Дубаков ВВЕДЕНИЕ В ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ НА JAVA**](https://dl.spbstu.ru/mod/resource/view.php?id=39178)**»**

Exception (Исключение) - это событие, которое происходит во время выполнения программы, нарушая нормальный ход выполнения ее команд. Обработка исключений - важная часть Java-программирования, которая позволяет помещать потенциально опасный код в блок try (что означает: "попробуем так и посмотрим, вызовет ли это исключение") и добавлять средства для обнаружения исключений (ошибок) различного типа.

Java предоставляет возможность использования блоков try - catch-finally, чтобы пытаться выявлять и программно обрабатывать ошибки времени исполнения.

Блоки try, catch и finally вместе образуют "капкан для ловли исключений". Вначале код, который является потенциально опасным и может вызвать исключение, заключается в состав оператора try. Если предполагаемое исключение происходит, управление сразу передается соответствующему блоку catch, который обрабатывает исключение. Когда один или оба оператора выполнены, управление передается блоку finally, который выполняется всегда, вне зависимости от того, имело место исключение или нет. "Поймав" исключение, можно попытаться аккуратно обойти его или же выйти из программы (или метода).

Язык Java поддерживает иерархию исключений, состоящую из большого количества типов исключений, сгруппированных в две основные категории: Контролируемые исключения проверяются компилятором (то есть компилятор проверяет, что ваш код где-то обрабатывает их). Неконтролируемые исключения (или исключения времени выполнения) не проверяются компилятором.

Error — это подкласс, который представляет серьезные проблемы, возникающие во время выполнения приложения. Большинство из этих ошибок сигнализируют о ненормальном ходе выполнения программы, т.е. о наступлении критических проблемах и эти ошибки не рекомендуется отмечать в методах посредством throws-объявления, и поэтому они называются не проверяемые (unchecked). Exception - это подкласс в иерархии, на который при программировании на Java следует уделять основное внимание. Этот уровень иерархии также разделяется на две ветви: исключения, производные от класса RuntimeException, и IOExeption.

Для определения списка исключений, которые могут выбрасываться методом, используется оператор throws. Если метод в явном виде (т.е. с помощью оператора throw) выбрасывает исключение соответствующего класса, тип класса исключений должен быть указан в операторе throws в объявлении этого метода.

Вообще говоря, в Java имеется пять ключевых операторов для работы с исключениями:

* try - данное ключевое слово используется для отметки начала блока кода, который потенциально может привести к ошибке.
* catch - ключевое слово для отметки начала блока кода, предназначенного для перехвата и обработки исключений.
* finally - ключевое слово для отметки начала блока кода, которое является дополнительным. Этот блок помещается после последнего блока catch. Управление обычно передаётся в блок finally в любом случае.
* throw - служит для генерации исключений. Оператор throw используется для возбуждения исключения «вручную».

throws - ключевое слово, которое прописывается в сигнатуре метода и обозначающее, что метод потенциально может выбросить исключение с указанным типом.