Министерство цифрового развития

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

Факультет Кибербезопасности и управления

Кафедра Управление в технических системах

**Отчёт**

**По лабораторной работе №2**

по дисциплине Прикладное программирование

**ВЫПОЛНИЛ**

студент ПрИ-22

Дубинкин Данила

**ПРОВЕРИЛ**

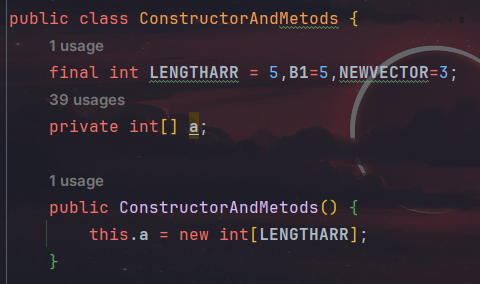
**Гильмуллин Рауль Ильнурович**

**Кадирова Валерия Анатольевна**

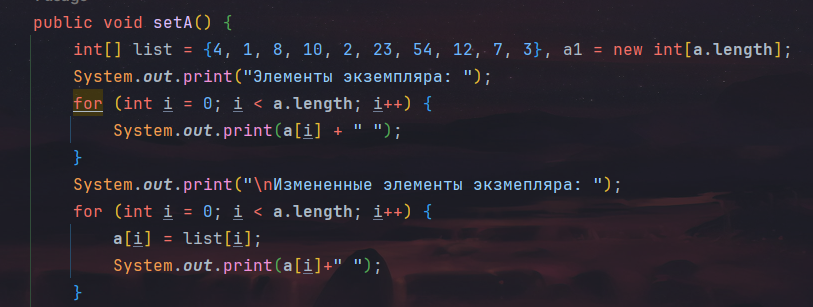
Самара

2024

Экземпляр должен соответствовать вектору фиксированной длины (она задается как параметр конструктора).



доступа к элементам вектора (получения значения и изменения значения)



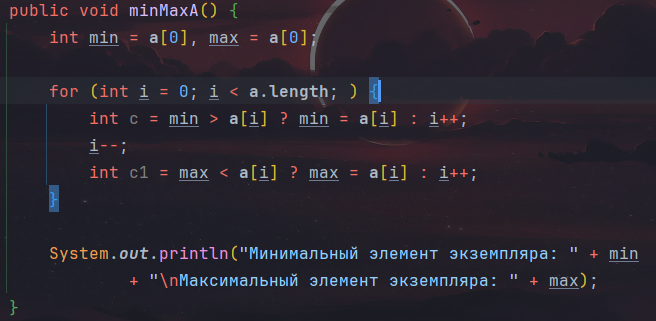


получения «длины» вектора (количества его элементов)



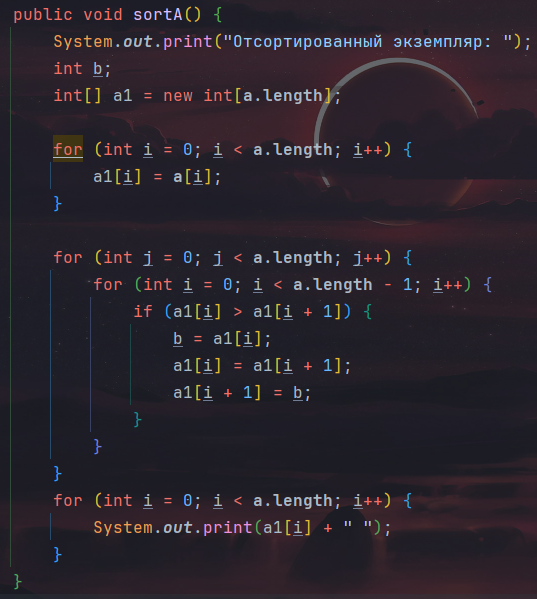


поиска минимального и максимального значений из элементов вектора



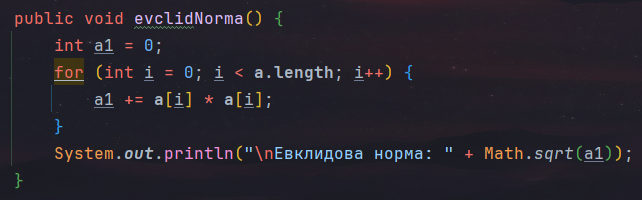


сортировки вектора (по возрастанию или убыванию – на ваш выбор)



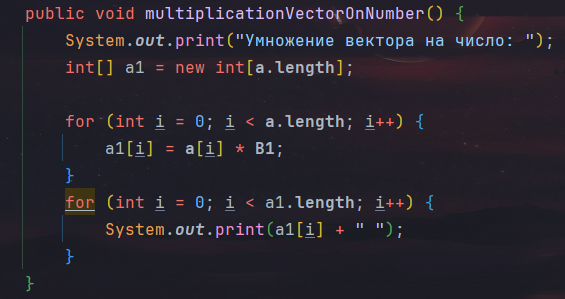


нахождения евклидовой нормы



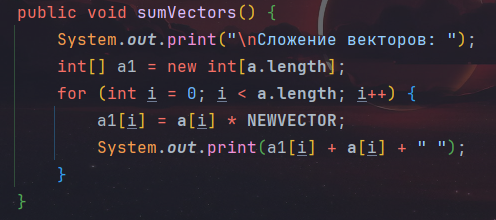


умножения вектора на число



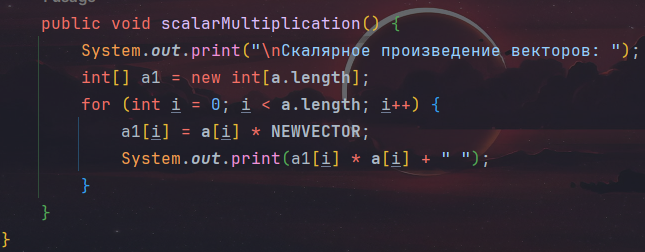


сложения двух векторов





нахождения скалярного произведения двух векторов





1. Принципы ООП: ООП (объектно-ориентированное программирование) включает в себя набор принципов, таких как инкапсуляция, наследование, полиморфизм и абстракция. В Java все сущности являются объектами, и они следуют этим принципам ООП.

2. Комментарии в Java: В Java существуют три вида комментариев: однострочные комментарии, многострочные комментарии и документационные комментарии. Однострочные комментарии начинаются с `//`, многострочные комментарии - с `/\*` и заканчиваются на `\*/`, а документационные комментарии - с `/\*\*` и также заканчиваются на `\*/`.

3. Идентификаторы и правила именования: Идентификатор в Java - это имя переменной, метода, класса и т. д. В Java название идентификаторов должно начинаться с буквы, символа подчеркивания `\_` или знака доллара `$`. После начального символа могут следовать буквы, цифры, символ подчеркивания или знак доллара.

4. Примитивные типы данных: В Java есть 8 примитивных типов данных: byte, short, int, long, float, double, boolean, char. Они предоставляют базовые данные для хранения чисел, символов и логических значений.

5. Понятие одномерного массива: Одномерный массив в Java является упорядоченной коллекцией элементов одного типа данных. Он обеспечивает эффективное хранение данных и обеспечивает простой доступ к этим данным. Для создания массива следует использовать синтаксис: `тип\_данных[] имя\_массива = new тип\_данных[размер];`.

1. Понятие многомерного массива: Многомерный массив в Java представляет собой массив массивов. Это может быть двумерный, трехмерный и т.д. Для создания многомерного массива в Java используется синтаксис: `тип\_данных[][] имя\_массива = new тип\_данных[размер1][размер2];` для двумерного массива.

7. Оператор if и switch:

- Оператор if используется для выполнения блока кода, если некоторое условие истинно. Например, это может быть использовано для выполнения различных действий в зависимости от введенного пользователем значения.

- Оператор switch используется для выбора выполнения кода в зависимости от значения выражения. Это особенно удобно, когда нужно выполнить различные действия в зависимости от значения конкретной переменной.

8. Циклы while и do while:

- Цикл while выполняет указанный блок кода до тех пор, пока условие истинно. Например, это может использоваться для чтения значений из ввода пользователя до тех пор, пока не будет введено определенное значение.

- Цикл do while сначала выполняет блок кода, а затем проверяет условие для продолжения выполнения. Он гарантирует, что блок кода будет выполнен хотя бы один раз.

9. Циклы for и улучшенный for:

- Цикл for используется для выполнения блока кода определенное количество раз. Это может использоваться, например, для перебора элементов массива или для выполнения серии операций.

- Улучшенный for, или for-each, используется для перебора элементов массива или коллекции без явного использования индексов.

10. Операторы break, continue, return:

- Оператор break используется для прекращения выполнения цикла или выбора в операторе switch.

- Оператор continue используется для пропуска текущей итерации цикла и перехода к следующей итерации.

- Оператор return используется для возврата значения из метода и завершения его выполнения.