

Министерство образования Республики Беларусь  
Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Методы трансляции

ОТЧЁТ  
к лабораторной работе №1  
на тему

**ОПРЕДЕЛЕНИЕ МОДЕЛИ ЯЗЫКА. ВЫБОР ИНСТРУМЕНТАЛЬНОЙ  
ЯЗЫКОВОЙ СРЕДЫ**

Выполнил: студент гр.253502 Дриневский К.В.

Проверил: ассистент кафедры информатики  
Гриценко Н.Ю.

Минск 2025

## СОДЕРЖАНИЕ

1 Цель работы.....	3
2 Подмножество языка программирования.....	4
2.1 Типы данных в языке программирования Java.....	4
2.2 Операторы циклов и условные операторы.....	4
2.3 Структуры данных и функции.....	5
3 Инструментальная языковая среда.....	6
Заключение.....	7
Список использованных источников.....	8
Приложение А(обязательное) Листинг программного кода.....	9

## 1 ЦЕЛЬ РАБОТЫ

Целью данной лабораторной работы является определение подмножества языка программирования, которое включает ключевые элементы, необходимые для разработки базовых программ. В подмножество должны быть включены числовые и текстовые константы, все типы переменных, операторы цикла, структуры данных, функции и условные операторы. Это подмножество должно быть достаточным для создания программ, демонстрирующих работу с основными конструкциями языка, и позволит изучить их взаимодействие в рамках конкретных задач.

Для выполнения работы необходимо также определить инструментальную языковую среду, включающую язык программирования, операционную систему и тип компьютера. Это позволит обеспечить корректную работу разработанных программ и продемонстрировать их функциональность. Выбор среды разработки должен учитывать поддержку всех необходимых элементов языка, таких как типы данных, операторы и структуры, а также удобство использования для реализации поставленных задач. Удобство разработки может включать наличие отладчиков, встроенных средств анализа кода и документации.

Таким образом, в рамках данной работы требуется разработать три программы, каждая из которых будет включать все элементы выбранного подмножества языка. Эти программы должны демонстрировать использование числовых и текстовых констант, работу с переменными различных типов, применение операторов цикла и условных операторов, а также использование структур данных и функций. Программы должны быть реализованы в выбранной инструментальной среде и корректно работать на указанной операционной системе. Дополнительно, необходимо провести ручное тестирование каждой программы, чтобы убедиться в их корректности и эффективности. Тестирование должно включать проверку всех возможных сценариев использования, а также оценку устойчивости программ к ошибкам. Важно также задокументировать результаты тестирования, чтобы иметь возможность в будущем анализировать найденные ошибки и улучшать код. Наконец, для успешного завершения лабораторной работы необходимо подготовить отчет, который будет содержать описание процесса разработки, выбор инструментальной среды, описание каждой программы и результаты тестирования.

## 2 ПОДМНОЖЕСТВО ЯЗЫКА ПРОГРАММИРОВАНИЯ

Язык программирования Java является универсальным инструментом для разработки, который поддерживает объектно-ориентированный подход с возможностью использования функциональных элементов. Java известен своей платформенной независимостью, безопасностью и богатой стандартной библиотекой. Синтаксис Java является строгим и статически типизированным, что позволяет создавать надежные и масштабируемые приложения. Этот язык идеально подходит для разработки корпоративных решений, мобильных приложений и веб-сервисов[1].

### 2.1 Типы данных в языке программирования Java

В Java существует два основных типа данных: примитивные и ссылочные.

Примитивные типы представляют собой простейшие формы данных, которые хранятся непосредственно в памяти. К ним относятся:

- byte. 8-битное целое число, диапазон от -128 до 127;
- short. 16-битное целое число, диапазон от -32,768 до 32,767;
- int. 32-битное целое число, диапазон от -2,147,483,648 до 2,147,483,647;
- long. 64-битное целое число, диапазон от -9,223,372,036,854,775,808 до 9,223,372,036,854,775,807;
- float. 32-битное число с плавающей точкой, используемое для представления дробных значений;
- double. 64-битное число с плавающей точкой, более точное, чем float;
- char. 16-битный символ, представляющий один символ Unicode;
- boolean. логический тип, принимающий значения true или false.

Ссылочные типы представляют собой объекты и массивы, которые хранят ссылки на данные, а не сами данные. Примеры включают:

- String. Последовательность символов, представляющая текст;
- Массивы. Коллекции фиксированного размера, например, int[] или String[].

### 2.2 Операторы циклов и условные операторы

Операторы циклов позволяют повторять блоки кода. В Java существуют три основных типа циклов:

- for используется для итерации по массивам или коллекциям;
- do...while выполняет блок кода хотя бы один раз и затем продолжает его выполнение, пока условие истинно;
- while проверяет условие и в случае его истинности выполняет блок кода;

Условные операторы позволяют выполнять разные действия в зависимости от условий:

- if...else проверяет условие и выполняет соответствующий блок кода;
- switch позволяет выбирать одно из множества значений.

## 2.3 Структуры данных и функции

Java предлагает множество структур данных для хранения и организации данных:

- массивы. Фиксированные по размеру структуры, которые могут хранить элементы одного типа.
- ArrayList. Динамический массив, который может изменять свой размер во время выполнения.
- HashMap. Структура данных, которая хранит пары "ключ-значение".
- HashSet. Коллекция, которая не допускает дубликатов и не гарантирует порядок элементов.

Функции в языке Java – это блоки кода, которые выполняют определенные задачи и могут быть вызваны из других частей программы. Они помогают организовать код, делают его более читаемым. В Java функции реализуются через методы. Методы могут быть статическими или нестатическими и могут принимать параметры и возвращать значения. Методы помогают структурировать код и повторно использовать его, что делает программу более удобной для чтения и сопровождения.

### 3 ИНСТРУМЕНТАЛЬНАЯ ЯЗЫКОВАЯ СРЕДА

Инструментальная языковая среда, используемая для разработки, основана на языке программирования C++23. C++ представляет собой современный язык, который сочетает в себе высокую производительность и гибкость написания кода. Это результат новейших исследований в области языков программирования, а также десятилетий опыта создания программного обеспечения, работающего на миллионах устройств. Именованные параметры и различные синтаксические конструкции делают C++ удобными для чтения и поддержки. Кроме того, разработчики могут использовать обширные возможности, предлагаемые стандартной библиотекой, что упрощает создание сложных приложений[2].

Разработка ведётся в операционной системе Windows 11 Home Single Language 24H2, которая предоставляет стабильную и надёжную платформу для программирования. Хотя C++ изначально разрабатывался для различных экосистем, его использование на Windows стало стандартом благодаря поддержке компиляторов и других инструментов. Windows 11 обеспечивает интеграцию с популярными редакторами кода, такими как Visual Studio Code, а также с системами управления версиями, что делает процесс разработки более комфортным и продуктивным. Кроме того, система предлагает доступ к богатому набору инструментов для тестирования и отладки, что особенно важно при создании качественного программного обеспечения.

Основным рабочим устройством является ноутбук Honor MagicBook 16. Этот ноутбук оснащён современным процессором Ryzen 5 5600H, который обеспечивает высокую производительность и позволяет работать с ресурсоёмкими задачами, такими как компиляция кода, тестирование приложений и работа с большими проектами. Большой объём оперативной памяти и быстрый накопитель обеспечивают комфортную работу даже при многозадачности. Надёжность и мобильность ноутбука делают его отличным выбором для выполнения сложных задач и продуктивной работы над проектами. Продуманная система охлаждения и качественная сборка обеспечивают стабильную работу даже при выполнении ресурсоёмких операций.

Таким образом, выбор инструментальной языковой среды для разработки на языке C++, в сочетании с использованием операционной системы Windows 11 и ноутбука Honor MagicBook 16, создаёт оптимальные условия для эффективного программирования. C++, как современный язык с высокой производительностью и гибкостью, предоставляет разработчикам все необходимые инструменты для реализации разнообразных проектов. Windows 11, благодаря своей стабильности и поддержке популярных инструментов, дополняет этот процесс, обеспечивая комфортную и продуктивную среду для работы.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы было проведено определение подмножества языка программирования Java, включая все основные элементы, необходимые для разработки программ. Java – это объектно-ориентированный язык, который предлагает разработчикам мощные инструменты для создания приложений. В рамках работы были изучены ключевые аспекты языка, такие как типы переменных, операторы циклов, структуры данных, функции и условные операторы. Это подмножество охватывает все стандартные возможности Java, что позволяет использовать его для решения широкого круга задач, начиная от простых арифметических вычислений и заканчивая сложной обработкой данных и созданием масштабируемых приложений.

Особое внимание было уделено выбору инструментальной среды. Выбранная среда разработки Visual Studio Code обеспечивает стабильную работу всех компонентов языка C++ и предоставляет удобные инструменты для написания, тестирования и отладки программ. Этот редактор поддерживает функциональные возможности языка, такие как автозаполнение кода, интеграцию с системами управления версиями и расширения для отладки, что значительно упрощает процесс разработки и повышает производительность.

Разработанные в рамках работы программы демонстрируют применение различных элементов подмножества языка Java. Они включают работу с числовыми и текстовыми константами, использование различных типов переменных, применение операторов циклов и условных операторов, а также манипуляции со структурами данных и функциями. Примеры программ иллюстрируют, как эти конструкции могут быть использованы для выполнения задач, таких как обработка списков, работа с коллекциями и реализация алгоритмов.

В заключение, выполнение данной лабораторной работы позволило не только изучить основные элементы языка Java, но и получить практический опыт их применения. Это подмножество языка, а также выбранная инструментальная среда, могут быть успешно использованы для решения задач в области анализа данных, разработки программного обеспечения, автоматизации бизнес-процессов и других направлений. Полученные знания и навыки станут прочной основой для дальнейшего изучения языка Java и его применения в профессиональной деятельности, открывая новые горизонты в мире программирования и разработки высококачественных приложений.

## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

- [1] Oracle [Электронный ресурс]. – Режим доступа: <https://docs.oracle.com/en/java/index.html>. – Дата доступа: 29.01.2025.
- [2] Cppreference [Электронный ресурс]. – Режим доступа: <https://en.cppreference.com/w/>. – Дата доступа: 29.01.2025.



## ПРИЛОЖЕНИЕ А

### (обязательное)

#### Листинг программного кода

##### Листинг 1 – DataTypesExample.java

```
public class DataTypesExample {
    public static void main(String[] args) {
        // Примитивные типы данных
        int age = 25;
        double salary = 55000.50;
        char grade = 'A';
        boolean isEmployed = true;

        // Условный оператор if
        if (isEmployed) {
            System.out.println("Возраст: " + age);
            System.out.println("Зарплата: $" + salary);
            System.out.println("Оценка: " + grade);
        } else {
            System.out.println("Не трудоустроен");
        }

        // Оператор switch case для оценки
        switch (grade) {
            case 'A':
                System.out.println("Отличная работа!");
                break;
            case 'F':
                System.out.println("Неудовлетворительно.");
                break;
            default:
                System.out.println("Недоступная оценка.");
                break;
        }
    }
}
```

##### Листинг 2 – LoopArrayExample.java

```
public class LoopArrayExample {
    public static void main(String[] args) {
        // Массив целых чисел
        int[] numbers = {1, 2, 3, 4, 5};

        // Цикл for
        System.out.print("Числа в массиве: ");
        for (int i = 0; i < numbers.length; i++) {
            System.out.print(numbers[i] + " ");
        }
        System.out.println();

        // Цикл do...while
        int index = 0;
        System.out.print("Числа в обратном порядке (do...while): ");
        do {
            System.out.print(numbers[numbers.length - 1 - index] + " ");
            index++;
        } while (index < numbers.length);
        System.out.println();

        // Цикл while
        int sum = 0;
```

```

        int j = 0;
        while (j < numbers.length) {
            sum += numbers[j];
            j++;
        }
        System.out.println("Сумма чисел в массиве: " + sum);
    }
}

```

### Листинг 3 – DataStructuresExample.java

```

import java.util.ArrayList;
import java.util.HashMap;

public class DataStructuresExample {
    // Функция для отображения элементов списка
    public static void displayList(ArrayList<String> list) {
        System.out.println("Элементы списка:");
        for (String item : list) {
            System.out.println(item);
        }
    }

    public static void main(String[] args) {
        // ArrayList
        ArrayList<String> fruits = new ArrayList<>();
        fruits.add("Яблоко");
        fruits.add("Банан");
        fruits.add("Вишня");

        // HashMap
        HashMap<String, Integer> fruitPrices = new HashMap<>();
        fruitPrices.put("Яблоко", 50);
        fruitPrices.put("Банан", 30);
        fruitPrices.put("Вишня", 80);

        // Вызов функции
        displayList(fruits);

        // Вывод цен на фрукты
        System.out.println("Цены на фрукты:");
        for (String fruit : fruitPrices.keySet()) {
            System.out.println(fruit + ": " + fruitPrices.get(fruit) + "
руб.");
        }
    }
}

```