

Plan

# 2023-09-13

- 3. Scanner
- 4. Workshop: if-else, if-else\_if-else, switch-case

- 3. Scanner
- 4. Практикум: if-else, if-else\_if-else, switch-case

Theory

## Scanner Theory

The Scanner class in Java is used for obtaining user input.

- Initializing a scanner: `Scanner scanner = new Scanner(System.in);`

Description	Method Call	Return Type	Example Usage	Example Console Input
Read a line	<code>scanner.nextLine()</code>	String	<code>String line = scanner.nextLine();</code>	my name is
Read a word	<code>scanner.next()</code>	String	<code>String word = scanner.next();</code>	Hello
Read an integer	<code>scanner.nextInt()</code>	int	<code>int number = scanner.nextInt();</code>	5
Read a double	<code>scanner.nextDouble()</code>	double	<code>double value = scanner.nextDouble();</code>	3.14
Read a boolean value	<code>scanner.nextBoolean()</code>	boolean	<code>boolean flag = scanner.nextBoolean();</code>	true

### next vs. nextLine

- **scanner.next():** This method reads a word from the input. If there are spaces in the input, **next()** will return only the first word until the nearest space. Spaces after that word will remain in the input queue. Example:

```
Scanner scanner = new Scanner(System.in);
String word = scanner.next();
System.out.println(word);
```

If you input "Hello World," word will contain "Hello."

- **scanner.nextLine()**: This method reads the entire line until the newline character ('\n'). This means it reads

```
Scanner scanner = new Scanner(System.in);
String line = scanner.nextLine();
System.out.println(line);
```

If you input "Hello World," **line** will contain "Hello World."

```
import java.util.Scanner;

public class ScannerExample {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter a number:");
        int x = scanner.nextInt();
        System.out.println("You entered: " + x);
    }
}
```

- If the user inputs an incorrect value, the behavior of the **nextInt()**, **nextDouble()**, and **nextBoolean()** methods will result in an exception (error).
- To avoid exceptions, it is recommended to perform checks before calling these methods to ensure that the user's input matches the expected format. You can use the **hasNextInt()**, **hasNextDouble()**, and **hasNextBoolean()** methods to verify whether it is safe to read a value of a specific type and take additional actions in case of incorrect input.

Description	Method Call	Return Type	Example Usage
Checking the presence of an integer	<code>scanner.hasNextInt()</code>	boolean	<code>if (scanner.hasNextInt()) { ... }</code>
Checking the presence of a double	<code>scanner.hasNextDouble()</code>	boolean	<code>if (scanner.hasNextDouble()) { ... }</code>

## Workshop (if-else, if-else\_if-else, switch-case)

### Using Different Types of switch-case

#### Key Points to Know

- Each **case** branch should end with a **break** statement to prevent the execution of other branches.
- **default** is a block that executes if none of the **case** options match the expression.

- The expression in a switch can be of different types: **integer** (`byte`, `short`, `int`, `long`), **strings** (`String`), **characters** (`char`), **enumerations** (`enum`), and even **objects**.
- It's essential to use the **break** statement to avoid "falling through" to other **case** options.
- If **break** is omitted, code execution will continue in the next **case**.

```
String choice = "tea"; // Let's assume the user chose tea
```

```
switch (choice) {  
    case "coffee":  
        System.out.println("You chose coffee.");  
        break;  
    case "tea":  
        System.out.println("You chose tea.");  
        break;  
    case "soda":  
        System.out.println("You chose soda.");  
        break;  
    default:  
        System.out.println("Invalid choice.");  
}
```

## Operators `&`, `&&`, `|`, and `||` in Java

- Operators `&`, `&&`, `|`, and `||` are used for performing logical operations.

### `&` (Bitwise AND) `&&` (Logical AND)

- `&`, `&&` perform a logical **AND** operation between two conditions.
- Returns true only if **both** conditions are true (equal to true).

### `|` (Bitwise OR) `||` (Logical OR)

- `|`, `||` perform a logical **OR** operation between two conditions.
- Returns true if **at least one** of the conditions is true (equal to true).

## Differences between `&` and `&&`, `|` and `||`

- `&&` and `||` have short-circuiting behavior. This means that if the result can be determined based on the first operand, the second operand is not evaluated.

## Tasks

- Write a program where the user first enters their first name, last name, and how to address them. After entering these three values, the program should greet the user.

Example:

Input in the console:

Andrej

Reutow

Herr

Output:

Hello, Herr Andre Reutow

- The user enters a number from 1 to 12, and the program tells which season corresponds to the entered month.

Example:

Input in the console:

7

Output:

You entered month 7, which is summer.

- The user enters a number from 1 to 12, and the program tells both the season and the name of the month corresponding to the entered number.

Example:

Input in the console:

8

Output:

You entered 8, August, which is summer.

- Write a program to calculate the payment to an employee with consideration of their years of service, assuming a base salary of \$500. For 3 years of service, there's a 10% bonus, for 5 years - a 50% bonus, for 10 years - a 100% bonus, and for 15 years - a 150% bonus. The number of years of service is entered by the user via the keyboard. (You can use if or switch at your discretion)
- Rewrite the tasks from the previous lecture's homework using Scanner.

## Scanner

### Теория

Класс Scanner в Java используется для получения ввода пользователя.

- подготовка сканера: `Scanner scanner = new Scanner(System.in);`

описание	вызов	возвращаемый тип	пример использования	пример ввода в консоль
чтение строки	<code>scanner.nextLine()</code>	String	<pre>String line = scanner.nextLine();</pre>	my name is
Чтение слова	<code>scanner.next()</code>	String	<pre>String word = scanner.next();</pre>	Hello
Чтение целого числа	<code>scanner.nextInt()</code>	int	<pre>int number = scanner.nextInt();</pre>	5

описание	вызов	возвращаемый тип	пример использования	пример ввода в консоль
Чтение числа с плавающей точкой	<code>scanner.nextDouble()</code>	<code>double</code>	<code>double value = scanner.nextDouble();</code>	3.14
Чтение логического значения	<code>scanner.nextBoolean()</code>	<code>boolean</code>	<code>boolean flag = scanner.nextBoolean();</code>	true

## next vs nextLine

- **scanner.next():** Этот метод читает слово из ввода. Если во вводе есть пробелы, то **next()** вернет только первое слово до ближайшего пробела. Пробелы после этого слова останутся во входной очереди. Пример:

```
Scanner scanner=new Scanner(System.in);
String word=scanner.next();
System.out.println(word);
```

Если вводите "Hello World", то word будет содержать "Hello".

- **scanner.nextLine():** Этот метод читает всю строку до символа новой строки ('\n'). Это означает, что он считает всю строку включая пробелы и другие символы, пока не встретит символ новой строки. Пример:

```
Scanner scanner = new Scanner(System.in);
String line = scanner.nextLine();
System.out.println(line);
```

Если вводите "Hello World", то line будет содержать "Hello World".

```
import java.util.Scanner;

public class ScannerExample {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter a number:");
        int x = scanner.nextInt();
        System.out.println("You entered: " + x);
    }
}
```

- Если пользователь вводит неверное значение, то поведение методов **nextInt()**, **nextDouble()**, и **nextBoolean()** вызовет исключение (ошибку)

- Чтобы избежать исключений, рекомендуется использовать проверки перед вызовом этих методов, чтобы убедиться, что ввод пользователя соответствует ожидаемому формату. Вы можете использовать методы **hasNextInt()**, **hasNextDouble()**, и **hasNextBoolean()**, чтобы проверить, можно ли безопасно прочитать значение определенного типа, и предпринять дополнительные действия в случае некорректного ввода.

описание	вызов	возвращаемый тип	пример использования
Проверка наличия целого числа	<code>scanner.hasNextInt()</code>	<code>boolean</code>	<pre>if (scanner.hasNextInt()) { ... }</pre>
Проверка наличия числа с плавающей точкой	<code>scanner.hasNextDouble()</code>	<code>boolean</code>	<pre>if (scanner.hasNextDouble()) { ... }</pre>

## Практикум (if-else, if-else\_if-else, switch-case)

### Использование разных типов switch-case

#### Важно знать

- Каждая ветвь **case** должна заканчиваться оператором `break`, чтобы предотвратить выполнение других ветвей.
- **default** - это блок, который выполняется, если ни один из вариантов `case` не соответствует выражению.
- Выражение в `switch` может быть различных типов: **целочисленные** (`byte`, `short`, `int`, `long`), **строки** (`String`), **символы** (`char`), **перечисления** (`enum`) и даже **объекты**.
- Необходимо использовать **break**, чтобы избежать "**проваливания**" в другие `case`.
- Если `break` пропущен, выполнение кода будет продолжено в следующем `case`.

```
String chose = "чай"; // Пусть пользователь выбрал чай
```

```
switch (chose) {
    case "кофе":
        System.out.println("Вы выбрали кофе.");
        break;
    case "чай":
        System.out.println("Вы выбрали чай.");
        break;
    case "газировка":
        System.out.println("Вы выбрали газировку.");
        break;
    default:
        System.out.println("Неверный выбор.");
}
```

# Операторы &, &&, | и || в Java

- операторы &, &&, | и ||, которые используются для выполнения логических операций.

## & (Побитовое И) && (Логическое И)

- &, && выполняет логическое **И** между двумя условиями.
- возвращает true только в том случае, если **оба** условия истинны (равны true).

## | (Побитовое ИЛИ) || (Логическое ИЛИ)

- |, || выполняет логическое **ИЛИ** между двумя условиями.
- возвращает true, если хотя бы **одно** из условий истинно (равно true).

## Различия между & и &&, | и ||

- && и || обладают короткой цепочкой вычислений. Это означает, что если значение результата можно определить на основе первого операнда, второй операнд не вычисляется.

## Задачи

- Написать программу, в которой пользователь вводит сначала свое имя, потом фамилия и как к нему обращаться. После ввода этих 3 значений программа должна поприветствовать пользователя.

Пример:

Ввод в консоль:

Andrej

Reutow

Herr

Вывод:

Здравствуйте, Herr Andre Reutow

- Пользователь с клавиатуры вводит число от 1 до 12, программа сообщает, какому времени года принадлежит введенный месяц.

Пример:

Ввод в консоль:

7

Вывод:

Вы ввели 7 месяц, это лето

- Пользователь с клавиатуры вводит число от 1 до 12, программа сообщает, какому времени года принадлежит введенный месяц и какой это месяц.

Пример:

Ввод в консоль:

8

Вывод:

Вы ввели 8, Август, это лето

- Напишите программу расчета выдачи денег работнику с учётом надбавки за стаж, если базовая зарплата 500\$. За 3 года работы надбавка 10%, за 5 лет работы надбавка 50%, за 10 лет работы надбавка 100%, за 15 лет работы надбавка 150%. Сколько лет проработал работник вводится с клавиатуры. (используйте на усмотрение if или switch)
- переписать задачи из дз прошлой лекции с использованием Scanner

## Homework

# Tasks

1. Understand the code in the LogicalOperator class.

- Why and in what case does the console output differ?

1. Write a program where the user first enters their first name, then last name, and how to address them. After entering these 3 values, the program should greet the user.

Example:

Console Input:

Andrej

Reutow

Herr

Output:

Hello, Herr Andre Reutow

3. The user enters a number from 1 to 12 from the keyboard, the program tells which season the entered month belongs to.

Example:

Console Input:

7

Output:

You entered the 7th month, it's summer

4. The user enters a number from 1 to 12 from the keyboard, the program tells which season the entered month belongs to and what month it is.

Example:

Console Input:

8



Output:

You entered 8, August, it's summer

5. Write a program to calculate the money paid to an employee, taking into account the bonus for experience, if the base salary is \$500. A 10% bonus for 3 years of work, a 50% bonus for 5 years of work, a 100% bonus for 10 years of work, a 150% bonus for 15 years of work. The number of years the employee has worked is entered from the keyboard. (Use either if or switch at your discretion)
6. Rewrite tasks from the homework of the last lecture using Scanner.

## Задачи:

1. разобраться с кодов в классе LogicalOperator.
  - почему и в каком случае вывод в консоль отличается?
2. Написать программу, в которой пользователь вводит сначала свое имя, потом фамилия и как к нему обращаться. После ввода этих 3 значений программа должна поприветствовать пользователя.

Пример:

Ввод в консоль:

Andrej  
Reutow  
Herr

Вывод:

Здравствуйτε, Herr Andre Reutow

3. Пользователь с клавиатуры вводит число от 1 до 12, программа сообщает, какому времени года принадлежит введенный месяц.

Пример:

Ввод в консоль:

7

Вывод:

Вы ввели 7 месяц, это лето

4. Пользователь с клавиатуры вводит число от 1 до 12, программа сообщает, какому времени года принадлежит введенный месяц и какой это месяц.

Пример:

Ввод в консоль:

8

Вывод:

Вы ввели 8, Август, это лето

5. Напишите программу расчета выдачи денег работнику с учётом надбавки за стаж, если базовая зарплата 500\$. За 3 года работы надбавка 10%, за 5 лет работы надбавка 50%, за 10 лет работы надбавка 100%, за 15 лет работы надбавка 150%. Сколько лет проработал работник вводится с клавиатуры. (используйте на усмотрение if или switch)
6. переписать задачи из дз прошлой лекции с использованием Scanner

## Code

code/HwSolution\_I06/src/AgeCalculator.java

```
// Напишите программу, которая принимает возраст
// и выводит соответствующую категорию (ребенок, подросток, взрослый)
public class AgeCalculator {
    public static void main(String[] args) {
        String result = calculateAgeCategory(7);
        System.out.println("Указан возраст 7 " + result);

        result = calculateAgeCategory(15);
        System.out.println("Указан возраст 15 " + result);

        result = calculateAgeCategory(150);
        System.out.println("Указан возраст 150 " + result);

        result = calculateAgeCategory(-1);
        System.out.println("Указан возраст -1 " + result);

        result = calculateAgeCategory(151);
        System.out.println("Указан возраст 151 " + result);
    }

    // Метод для определения категории возраста
    public static String calculateAgeCategory(int age) {
        String result; // ребенок 0 - 14, подросток 15 - 20, взрослый 21 - :
        boolean isKind = age >= 0 & age <= 14;

        if (isKind) {
            result = "ребенок";
        } else if (age >= 15 && age <= 20) {
            result = "подросток";
        } else if (age >= 21 && age <= 150) {
            result = "взрослый";
        }
    }
}
```

```
    } else {  
        result = "Такого возраста не существует";  
    }  
  
    return result;  
}  
}
```

code/HwSolution\_l06/src/Calculator.java

//Напишите программу, которая принимает два числа и операцию, а затем выпол

```
public class Calculator {  
    public static void main(String[] args) {  
        double result = performOperation(5, 5, '+');  
        System.out.println("Результат: " + result); // 10  
  
        result = performOperation(15, 5, '-');  
        System.out.println("Результат: " + result); // 10  
  
        result = performOperation(150, 100, 'A');  
        System.out.println("Результат: " + result); // 0  
    }  
  
    // Метод для выполнения арифметических операций  
    /*  
    char operator = '+' '-' '*' ...  
    */  
    public static double performOperation(double num1,  
                                           double num2,  
                                           char operator) {  
  
        double result = 0;  
        switch (operator) {  
            case '+':  
                result = num1 + num2;  
                break;  
            case '-':  
                result = num1 - num2;  
                break;  
            case '*':  
                result = num1 * num2;  
                break;  
            case '/':  
                result = num1 / num2;  
            }  
    }  
}
```

```
        break;
    case '%':
        result = num1 % num2;
        break;
    default:
        result = 0;
    }

    return result;
}
}
```

code/HwSolution\_l06/src/MoodEvaluator.java

//Напишите программу, которая принимает день недели и выводит ваше настроен

```
public class MoodEvaluator {
    public static void main(String[] args) {
        String result = evaluateMood("понедельник");
        System.out.println(result); // Фиговое

        result = evaluateMood("Вторник");
        System.out.println(result); // Чуть лучше чем в понедельник

        result = evaluateMood("вторник");
        System.out.println(result); // Такого дня недели не существует

        result = evaluateMood("пятница");
        System.out.println(result); // Такого дня недели не существует
    }
```

// Метод для оценки настроения в зависимости от дня недели

```
public static String evaluateMood(String dayOfWeek) {
    String result;

    switch (dayOfWeek) {
        case "понедельник":
            result = "Фиговое";
            break;
        case "Вторник":
            result = "Чуть лучше чем в понедельник";
    }
```

```
        break;
    default:
        result = "Такого дня недели не существует";
    }

    return result;
}
}
```

code/HwSolution\_l06/src/SeasonDeterminer.java

// Напишите программу, которая принимает месяц и выводит соответствующий се:

```
public class SeasonDeterminer {
    public static void main(String[] args) {
        String result = determineSeason(1);
        System.out.println(result); // Зима

        result = determineSeason(6);
        System.out.println(result); // Лето

        result = determineSeason(0);
        System.out.println(result); // Такого месяца не существует

        result = determineSeason(13);
        System.out.println(result); // Такого месяца не существует
    }

    // Метод для определения сезона по номеру месяца
    public static String determineSeason(int month) {
        String result;

        switch (month) {
            case 1:
            case 2:
            case 3:
            case 10:
            case 11:
            case 12:
                result = "Зима";
                break;
            case 4:
            case 5:
```

```
        result = "Весна";
        break;
    case 6:
    case 7:
    case 8:
        result = "Лето";
        break;
    case 9:
        result = "Осень";
        break;
    default:
        result = "Такого месяца не существует";
    }

    return result;
}
}
```

code/Lesson\_07/src/ScannerExample.java

```
import java.util.Scanner;

/**
 * @author Andrej Reutow
 * created on 13.09.2023
 */
public class ScannerExample {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Введите целое число");
        int number = scanner.nextInt();
        System.out.println("Введено число " + number);

        scanner.nextLine();

        System.out.println("Введите город в котором вы живете ?");
        String line = scanner.nextLine();
        System.out.println("Вы живете в городе " + line);
    }
}
```

code/Lesson\_07/src/ScannerExampleExceptionHandle.java

```
import java.util.Scanner;

/**
 * @author Andrej Reutow
 * created on 12.09.2023
 */
public class ScannerExampleExceptionHandle {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Проверяем, можно ли считать целое число
        if (scanner.hasNextInt()) {
            int intValue = scanner.nextInt();
            System.out.println("Вы ввели целое число: " + intValue);
        } else {
            System.out.println("Это не целое число.");
            // Обработка некорректного ввода, например, запрос повторного в
        }

        scanner.nextLine();

        // Проверяем, можно ли считать число с плавающей точкой
        if (scanner.hasNextDouble()) {
            double doubleValue = scanner.nextDouble();
            System.out.println("Вы ввели число с плавающей точкой: " + doubleValue);
        } else {
            System.out.println("Это не число с плавающей точкой.");
            // Обработка некорректного ввода, если необходимо.
        }

        // Закрываем сканнер
        scanner.close();
    }
}
```

code/Lesson\_07/src/LogicalOperator.java

```
/**
 * @author Andrej Reutow
 * created on 13.09.2023
 */
```

```
public class LogicalOperator {
    public static void main(String[] args) {
        // true && true
        if (checkAge(19) || checkTicket(true)) {
            System.out.println("Person is year old and has ticket true");
        }

        // false && false
        if (checkAge(10) || checkTicket(false)) {
            System.out.println("Person is 10 year old and has ticket false")
        }

        // false && true
        if (checkAge(10) || checkTicket(true)) {
            System.out.println("Person is 10 year old and has ticket true")
        }

        // true && false
        if (checkAge(30) || checkTicket(false)) {
            System.out.println("Person is 30 year old and has ticket false")
        }
    }

    public static boolean checkAge(int age) {
        System.out.println("Check age");
        if (age >= 18) {
            return true;
        } else {
            return false;
        }
    }

    public static boolean checkTicket(boolean hasTicket) {
        System.out.println("Check ticket");
        if (hasTicket) {
            return true;
        } else {
            return false;
        }
    }
}
```