

## Plan

# 2023-09-14

2. Repetition and practice

3. Повторение и практика

## Theory

**Для того чтобы успевать за нами писать код, вы должны найти на своей клавиатуре следующие символы**

# < > + - , . : ; ( ) { } / \

## Ключевые слова Всего зарезервировано 50 служебных (ключевых) слов в Java.

Список ключевых слов по алфавиту:

- abstract assert boolean break byte case catch char class const continue default do double else enum extends final finally float for goto if implements import instanceof int interface long native new package private protected public return short static strictfp super switch synchronized this throw throws transient try void volatile while

## Операторы сравнения

- Это как в школе: больше, меньше, равно.

> больше < меньше == равно != не равно >= больше или равно <= меньше или равно

```
int a = 5;
```

```
int b = 10;
```

```
if (a > b) {
```

```
    // а больше b, но это неправда
```

```
}
```

**Заполните эту форму для закрепления** - <https://forms.gle/bh7bPSFDmdehLjfT7>

## Логические операторы

Это как соединение двух предложений: "и", "или", "не".

&& и || или ! не

```
if (a > 0 && b > 0) {  
    // а и b больше нуля  
}
```

## If-else

Это как сказать: "Если пойдёт дождь, возьму зонт. Иначе — кепку".

```
if (a > b) {  
    // Если a больше b  
} else {  
    // Если не выполнено условие a > b  
}
```

## Switch-case

Это как расписание уроков: "Если понедельник, иду на математику, если вторник — на физику".

```
int day = 1;  
  
switch (day) {  
    case 1:  
        System.out.println("Понедельник");  
        break;  
    case 2:  
        System.out.println("Вторник");  
        break;  
    default:  
        System.out.println("Неизвестный день");  
}
```

**На примере лифта**, спасибо Vitalij за пример

Представим лифт в многоэтажном здании. Каждая кнопка внутри лифта — это case в операторе switch.

При нажатии на кнопку с номером этажа, лифт "знает", куда ехать. Это можно представить с помощью switch-case в Java:

```
int этаж = 5; // Этаж, на который хотим поехать  
  
switch (этаж) {  
    case 1:  
        System.out.println("Едем на первый этаж");  
        break;  
    case 2:  
        System.out.println("Едем на второй этаж");  
        break;  
    case 3:  
        System.out.println("Едем на третий этаж");  
}
```

```
        break;
    case 4:
        System.out.println("Едем на четвертый этаж");
        break;
    case 5:
        System.out.println("Едем на пятый этаж");
        break;
    default:
        System.out.println("Такого этажа нет в этом здании");
}
```

- Каждый case — это кнопка с номером этажа.
- break — это команда лифту остановиться на выбранном этаже и не продолжать движение.
- default — это ситуация, когда кто-то нажал кнопку, которой нет в лифте (например, этаж "13", если его нет в здании).
- Таким образом, switch-case работает как пульт управления лифтом: выбираем нужный "случай" (case) и выполняем соответствующие действия.

## Разбор задачи CriptoCalculator:

### Объяснение и формулы:

### 1. На какую сумму Вася может рассчитывать, если продаст все свои монеты?

#### Формула:

$\text{Total Value} = \text{Initial Coins} * \text{Current Price}$

#### Объяснение:

Чтобы узнать, на какую сумму Вася может рассчитывать, нужно умножить текущее количество монет на текущую цену одной монеты.

---

### 2. Какая прибыль будет у Васи, если он продаст монеты сейчас?

#### Формула:

$\text{Profit} = (\text{Current Price} - \text{Initial Price}) * \text{Initial Coins}$

#### Объяснение:

Прибыль рассчитывается как разница между текущей и начальной ценой монеты, умноженная на количество монет, которые у Васи были изначально.

---

### 3. Сколько монет нужно продать, чтобы получить x прибыль?

**Формула:**

$\text{Coins for } x \text{ Profit} = \text{Desired Profit} / (\text{Current Price} - \text{Initial Price})$

**Объяснение:**

Чтобы узнать, сколько монет нужно продать для получения желаемой прибыли, нужно разделить желаемую прибыль на разницу между текущей и начальной ценой монеты.

## 4. Какая стоимость монеты должна быть, чтобы получить x прибыль?

**Формула:**

$\text{Price for } x \text{ Profit} = (\text{Desired Profit} / \text{Initial Coins}) + \text{Initial Price}$

**Объяснение:**

Чтобы узнать, какой должна быть стоимость монеты для получения желаемой прибыли, нужно разделить желаемую прибыль на изначальное количество монет и прибавить к этому начальную цену монеты.

## 5. Сколько монет нужно по текущему курсу, чтобы получить x сумму?

**Формула:**

$\text{Coins for } x \text{ Amount} = \text{Desired Amount} / \text{Current Price}$

**Объяснение:**

Чтобы узнать, сколько монет нужно для получения желаемой суммы, нужно разделить желаемую сумму на текущую цену одной монеты.

## 6. Всё то же самое, только любой доход облагается налогом. Налог 23%.

**Формула:**

$\text{Income after Tax} = \text{Income} * 0.77$

**Объяснение:**

Чтобы узнать доход после уплаты налогов, нужно умножить доход на 0.77 (так как налог составляет 23%, и остается  $100\% - 23\% = 77\%$ ).

Homework

[Code](#)

code/HwSolution\_I04/src/CallCost.java

```
public class CallCost {
    // Метод для расчета стоимости звонков
    public static double calculateCallCost(int minutes, double rate) {
        return minutes * rate;
    }

    public static void main(String[] args) {
        // Примеры использования метода для расчета стоимости звонков
        System.out.println("Стоимость 15 минут: " + calculateCallCost(15, 0.1));
        System.out.println("Стоимость 30 минут: " + calculateCallCost(30, 0.1));
    }
}
```

code/HwSolution\_I04/src/CalorieCalculation.java

```
public class CalorieCalculation {

    // Метод для расчета сожженных калорий
    public static int calculateCaloriesBurnt(int minutes, int caloriesPerMinute) {
        return minutes * caloriesPerMinute;
    }

    public static void main(String[] args) {
        // Примеры использования метода для расчета сожженных калорий
        System.out.println("Сожженные калории за 20 минут: " + calculateCaloriesBurnt(20, 10));
        System.out.println("Сожженные калории за 30 минут: " + calculateCaloriesBurnt(30, 10));
    }
}
```

code/HwSolution\_I04/src/EnergyCalculator.java

```
public class EnergyCalculator {

    // Расчет электроэнергии в час
    public static double calculateHourlyConsumption(double fridgeWatt, double acWatt) {
        return (fridgeWatt + acWatt) / 1000; // В кВт
    }

    // Расчет месячных расходов на электроэнергию
    public static double calculateMonthlyCost(double hourlyConsumption, double pricePerKwh) {
        return hourlyConsumption * 24 * 30 * pricePerKwh;
    }
}
```

```
        return hourlyConsumption * 24 * 30 * costPerKWh;
    }

    public static void main(String[] args) {
        // Примеры
        System.out.println("Месячные расходы (150 Вт холодильник, 2000 Вт к
            calculateMonthlyCost(calculateHourlyConsumption(150, 2000),
        System.out.println("Месячные расходы (200 Вт холодильник, 1800 Вт к
            calculateMonthlyCost(calculateHourlyConsumption(200, 1800),
    }
}
```

code/HwSolution\_l04/src/FuelCalculation.java

```
public class FuelCalculation {

    // Метод для расчета расхода топлива
    public static double calculateFuelConsumption(double distance, double f
        return (distance / 100) * fuelRate;
    }

    public static void main(String[] args) {
        // Примеры использования метода для расчета расхода топлива
        System.out.println("Топливо для 150 км: " + calculateFuelConsumptio
        System.out.println("Топливо для 300 км: " + calculateFuelConsumptio
    }
}
```

code/HwSolution\_l04/src/CryptoCalculator.java

```
public class CryptoCalculator {

    // 1. На какую сумму Вася может рассчитывать, если продаст все свои мон
    public static double calculateTotalValue(double initialCoins, double cu
        return initialCoins * currentPrice;
    }

    // 2. Какая прибыль будет у Васи, если он продаст монеты сейчас?
    public static double calculateProfit(double initialCoins, double initia
        return (currentPrice - initialPrice) * initialCoins;
    }
}
```

```
// 3. Сколько монет нужно продать, чтобы получить x прибыль?
public static double coinsForXProfit(double initialCoins, double initialPrice, double
    return desiredProfit / (currentPrice - initialPrice);
}

// 4. Какая стоимость монеты должна быть, чтобы получить x прибыль?
public static double priceForXProfit(double initialCoins, double initialPrice, double
    return (desiredProfit / initialCoins) + initialPrice;
}

// 5. Сколько монет нужно по текущему курсу, чтобы получить x сумму?
public static double coinsForXAmount(double currentPrice, double desiredAmount, double
    return desiredAmount / currentPrice;
}

// 6. Всё то же самое, только любой доход облагается налогом. Налог 23%
public static double applyTax(double income) {
    return income * 0.77; // 100% - 23% = 77%
}

public static void main(String[] args) {
    double initialCoins = 12350;
    double initialPrice = 0.27;
    double currentPrice = 16.38;

    System.out.println("Total Value: " + calculateTotalValue(initialCoins, initialPrice, currentPrice));
    System.out.println("Profit: " + calculateProfit(initialCoins, initialPrice, currentPrice));
    System.out.println("Coins for 1000 EUR Profit: " + coinsForXProfit(1000, initialPrice, currentPrice));
    System.out.println("Price for 1000 EUR Profit: " + priceForXProfit(1000, initialCoins, currentPrice));
    System.out.println("Coins for 1000 EUR Amount: " + coinsForXAmount(1000, currentPrice));
    System.out.println("Profit after Tax: " + applyTax(calculateProfit(initialCoins, initialPrice, currentPrice)));
}
}
```

code/Lesson\_08/src/Main.java

```
/**
 * @author Andrej Reutow
 * created on 14.09.2023
 */
public class Main {
    public static void main(String[] args) {
        System.out.println("Hello world!");
    }
}
```

```
}  
}
```

code/Lesson\_08/src/TasksLesson08.java

```
/**  
 * @author Andrej Reutow  
 * created on 14.09.2023  
 */  
public class TasksLesson08 {  
  
    public static void main(String[] args) {  
        System.out.println("- Task 1:");  
        // позитивные случаи  
        task1(1);        // Эконом  
        task1(500);       // Стандарт  
        task1(3000);      // Премиум  
        // негативные случаи  
        task1(-1);        // для вашего запроса нет тарифа  
        task1(0);         // для вашего запроса нет тарифа  
        System.out.println("=====");  
        System.out.println("- Task 2:");  
        // позитивные случаи  
        task2(1);          // ночь  
        task2(12);         // утро  
        task2(21);         // вечер  
        // негативные случаи  
        task2(25);          // сутки 24 часа, от 0 - до 23  
        task2(-5);          // сутки 24 часа, от 0 - до 23  
        System.out.println("=====FINISH=====");  
    }  
  
    /**  
    Определение тарифа для мобильной связи (if-else-if-else) Оператор мобил  
    три тарифа: "Эконом", "Стандарт" и "Премиум".  
    Если клиент использует менее 500 минут в месяц, ему подходит "Эконом".  
    Если от 500 до 1000 минут – "Стандарт". Если больше 1000 минут – "Преми  
    Программа должна принимать количество минут и выводить подходящий тариф  
    */  
    public static void task1(int minMinutes) {  
  
        if (minMinutes > 0 && minMinutes < 500) {  
            System.out.println("Эконом");  
        } else if (minMinutes >= 500 && minMinutes <= 1000) {
```



```
        System.out.println("Стандарт");
    } else if (minMinutes > 1000) {
        System.out.println("Премиум");
    } else {
        System.out.println("для вашего запроса нет тарифа");
    }
}

/*
Определение времени суток (if-else-if-else) Напишите программу,
которая принимает текущий час (от 0 до 23) и выводит приветствие в зави
от времени суток(утро, день, вечер, ночь).
*/
public static void task2(int hour) {

    // if (minMinute >= 500 && minMinute <= 1000)
    // если текущий час больше или равен 5
    // if hour >= 5
    // и текущий час меньше или равен 12
    if (hour >= 0 && hour <= 23) {
        // 5 .. 12
        if (hour >= 5 && hour <= 12) {
            System.out.println("утро");
        }

        // 13 ... 16
        if (hour >= 13 && hour <= 16) {
            System.out.println("день");
        }

        // 17 ... 21
        if (hour >= 17 && hour <= 21) {
            System.out.println("вечер");
        }

        // 22, 23 ...или 0 ... 4
        if ((hour == 22 || hour == 23) || (hour >= 0 && hour <= 4)) {
            System.out.println("ночь");
        }
    } else {
        System.out.println("сутки 24 часа, от 0 - до 23");
    }
}
```

```
//      if (hour >= 0 && hour <= 23) {  
//  
//          if (hour >= 5 && hour <= 12) {  
//              System.out.println("утро");  
//          } else if (hour >= 13 && hour <= 16) {  
//              System.out.println("день");  
//          } else if (hour >= 17 && hour <= 21) {  
//              System.out.println("вечер");  
//          } else {  
//              System.out.println("ночь");  
//          }  
//  
//      } else {  
//          System.out.println("сутки 24 часа, от 0 - до 23");  
//      }  
}
```