



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
"МИРЭА - Российский технологический университет"

**РТУ МИРЭА**

---

Институт информационных технологий (ИТ)  
Кафедра инструментального и прикладного программного обеспечения

**ОТЧЁТ  
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2  
по дисциплине  
«Программирование на языке Java»**

**Тема: ООП Java. Понятие класса**

Выполнил студент группы ИКБО-16-20

Пак С.А.

Принял ассистент кафедры ИиППО

Русляков А.А.

Практические работы выполнены «\_\_\_\_\_» 2021г.

«Зачтено» «\_\_\_\_\_» 2021г

Москва 2021

## СОДЕРЖАНИЕ

ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ.....	3
1. ООП в языке Java.....	3
2. Создание классов в Java.....	3
3. Создание экземпляра класса.....	3
РЕШЕНИЕ ЗАДАЧИ.....	4
1. Постановка задачи.....	4
2. Программный код.....	4
3. Вывод программы.....	6
ВЫВОД.....	8

# ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ

## 1. ООП в языке Java

Язык Java - объектно-ориентированный язык программирования. В центре ООП находится понятие объекта. Объект — это сущность, которой можно посылать сообщения и которая может на них реагировать, используя свои данные. Объект — это экземпляр класса. Данные объекта скрыты от остальной программы. Соккрытие данных называется инкапсуляцией. Наличие инкапсуляции достаточно для объектности языка программирования, но ещё не означает его объектной ориентированности — для этого требуется наличие наследования. Но даже наличие инкапсуляции и наследования не делает язык программирования в полной мере объектным с точки зрения ООП. Основные преимущества ООП проявляются только в том случае, когда в языке программирования реализован полиморфизм подтипов — возможность единообразно обрабатывать объекты с различной реализацией при условии наличия общего интерфейса.

## 2. Создание классов в Java

Синтаксис написания класса:

```
<модификатор_доступа> class <имя_класса> {  
    <тело_класса>  
}
```

В качестве модификатора доступа можно указать ключевое слово `public` или `private`. Если указано слово `public`, то класс будет доступен из других пакетов. Если указано слово `private`, то класс будет доступен только внутри того пакета, в котором он находится. В теле класса можно описать методы, переменные, константы, конструкторы класса. Конструктор - это специальный метод, который вызывается при создании нового объекта. Не всегда удобно инициализировать все переменные класса при создании его экземпляра. Иногда проще, чтобы какие-то значения были бы созданы по умолчанию при создании объекта. По сути конструктор нужен для автоматической инициализации переменных.

## 3. Создание экземпляра класса

Для того чтобы создать экземпляр класса необходимо объявить переменную, тип которой соответствует имени класса или суперкласса. После чего нужно присвоить этой переменной значение, вызвав конструктор создаваемого класса с помощью оператора `new`. Например, можно создать экземпляр класса `Rectangle` следующим образом:

```
Rectangle rect = new Rectangle(20, 10);
```

# РЕШЕНИЕ ЗАДАЧИ

## 1. Постановка задачи

**Вариант:** 3

**Задание:** создать класс, описывающий книгу (Book). В классе должны быть описаны нужные свойства книги(автор, название, год написания и т. д.)и методы для получения, изменения этих свойств. Протестировать работу класса в классе BookTest, содержащим метод статический main(String[] args).

## 2. Программный код

Файл Book.java:

```
package ru.mirea.classes;

public class Book {
    private String title;
    private String author;
    private int year;

    /**
     * Конструктор по умолчанию
     */
    public Book() {
        this.title = "Unnamed Book";
        this.author = "Anonymous";
        this.year = 0;
    }

    /**
     * Конструктор, создающий полноценную книгу
     * @param title      название книги
     * @param author     автор книги
     * @param date       год выхода книги
     */
    public Book(String title, String author, int year) {
        this.title = title;
        this.author = author;
        this.year = year;
    }

    /**
     * Возвращает название книги
     */
    public String getTitle() {
        return this.title;
    }
}
```

```

    }

    /**
     * Изменяет значение поля title
     * @param newTitle        новое название книги
     */
    public void setTitle(String newTitle) {
        this.title = newTitle;
    }

    /**
     * Возвращает имя автора книги
     */
    public String getAuthor() {
        return this.author;
    }

    /**
     * Изменяет значение поля author
     * @param newAuthor        новый автор книги
     */
    public void setAuthor(String newAuthor) {
        this.author = newAuthor;
    }

    /**
     * Возвращает год выхода книги
     */
    public int getYear() {
        return this.year;
    }

    /**
     * Изменяет значение поля year
     * @param year            новая год выхода книги
     */
    public void setYear(int newYear) {
        this.year = newYear;
    }
}

```

Файл BookTest.java:

```

package ru.mirea;

import java.util.Scanner;
import ru.mirea.classes.Book;

```

```

public class BookTest {
    private static final Scanner IN = new Scanner(System.in);

    public static void main(String[] args) {
        System.out.print("Введите название книги: ");
        String bookTitle = IN.nextLine();

        System.out.print("Введите имя автора книги: ");
        String bookAuthor = IN.nextLine();

        System.out.print("Введите год выхода книги: ");
        int bookYear = IN.nextInt();

        Book book = new Book();

        book.setTitle(bookTitle);
        book.setAuthor(bookAuthor);
        book.setYear(bookYear);

        System.out.println();
        System.out.println("Автором книги " + book.getTitle() + ",
которая вышла в " + book.getYear() + ", является " +
book.getAuthor());
    }
}

```

### 3. Вывод программы

Пользователь вводит 3 параметра: название книги, имя автора книги и год издания книги (рис.1, 2).

```

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

ezhik@spark:Java-MIREA-lab-2/out $ java ru.mirea.BookTest
Введите название книги: some book
Введите имя автора книги: some author
Введите год выхода книги: 1999

Автором книги some book, которая вышла в 1999, является some author
ezhik@spark:Java-MIREA-lab-2/out $ █

```

Рис.1 Выходные данные программы ч.1

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

ezhik@spark:Java-MIREA-lab-2/out $ java ru.mirea.BookTest
Введите название книги: Alice in Wonderland
Введите имя автора книги: Lewis Carroll
Введите год выхода книги: 1990

Автором книги Alice in Wonderland, которая вышла в 1990, является Lewis Carroll
ezhik@spark:Java-MIREA-lab-2/out $ █
```

Рис.2 Выходные данные программы ч.2

## **ВЫВОД**

В ходе выполнения изучил основные концепции объектно-ориентированного программирования, изучил понятие класса и научился создавать классы.