

4. Порядок инициализации статических и не статических переменных при наследовании

Для случая наследования классов:

```
public class Child extends Parent { }
```

поля объекта инициализируются в следующем порядке:

1. Статические поля класса `Parent`;
2. Статический блок инициализации класса `Parent`;
3. Статические поля класса `Child`;
4. Статический блок инициализации класса `Child`;
5. Нестатические поля класса `Parent`;
6. Нестатический блок инициализации класса `Parent`;
7. Конструктор класса `Parent`;
8. Нестатические поля класса `Child`;
9. Нестатический блок инициализации класса `Child`;
10. Конструктор класса `Child`.

Пример

```
import java.util.ArrayList;

public class Cat {

    // Переменные экземпляра и класса.
    private static ArrayList<String> order = new ArrayList<>();

    private Initializer initializer = new Initializer();
    private String name;
    private static String color;
    private int age;

    // 1. Статический инициализатор
    static {
        color = "Striped";
        order.add("static initializer");
    }

    // 2. Внутренний класс
    private class Initializer {
        Initializer() {
            order.add("class Initializer");
        }
    }

    // 3. Инициализатор
    {
        name = "Angry cat";
        order.add("non-static initializer");
    }

    // 4. Конструктор
    public Cat() {
        age = 10;
        order.add("Constructor");
    }

    // Геттер
    public static ArrayList<String> getOrder() {
        return order;
    }

    // Запуск
    public static void main(String[] args) {
        for (String s : new Cat().getOrder())
            System.out.println(s);
    }
}
```

Вывод

```
static initializer  
class Initializer  
non-static initializer  
Constructor
```