

Типы переменных в Java



ПРЕПОДАВАТЕЛЬ



Артем Гордийчук

Full-stack software engineer

- Более 8 лет опыта работы
- Java, Spring, Hibernate, AWS, Oracle, PostgreSQL
- Проекты связанные с банковской, финансовой деятельностью, e-commerce

artemsgor@gmail.com

www.linkedin.com/in/artem-g-48071a61



ВАЖНО:

- Камера должна быть включена на протяжении всего занятия.
- Если у Вас возник вопрос в процессе занятия, пожалуйста, поднимите руку и дождитесь, пока преподаватель закончит мысль и спросит Вас, также можно задать вопрос в чате или когда преподаватель скажет, что начался блок вопросов.
- Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях.
- Вести себя уважительно и этично по отношению к остальным участникам занятия.
- Во время занятия будут интерактивные задания, будьте готовы включить камеру или демонстрацию экрана по просьбе преподавателя.

ПЛАН ЗАНЯТИЯ

1. Повторение
2. Вопросы по повторению
3. Основной блок
4. Вопросы по основному блоку
5. Задание для закрепления
6. Практическая работа
7. Оставшиеся вопросы



TEL-RAN
by Starta Institute

1

ПОВТОРЕНИЕ ИЗУЧЕННОГО

Повторение

- Проект
- Пакет
- Класс
- Метод





TEL-RAN
by Starta Institute

2

ВОПРОСЫ ПО ПОВТОРЕНИЮ

Введение

- Переменные в Java
- Типы переменных
- Локальные переменные
- Переменные экземпляра
- Статические переменные
- Сравнение переменных
- Примеры типов переменных
- Сканер





TEL-RAN
by Starta Institute

3


ОСНОВНОЙ БЛОК

Переменные в Java

Переменная в Java — это контейнер данных, который сохраняет значения данных во время выполнения программы Java.

Каждой переменной назначается тип данных, определяющий тип и количество значений, которые она может содержать.

Переменная — это имя, присвоенное ячейке памяти.



Типы переменных в Java

Локальные переменные

Переменные экземпляра

Статические переменные



Локальные переменные

- Переменная, определенная в блоке, методе или конструкторе, называется локальной переменной.
- Эти переменные создаются при входе в блок или при вызове функции и уничтожаются после выхода из блока или при возврате вызова из функции.
- Область действия этих переменных существует только внутри блока, в котором переменные объявлены, т. е. мы можем получить доступ к этим переменным только внутри этого блока.
- Инициализация локальной переменной обязательна перед ее использованием в определенной области.



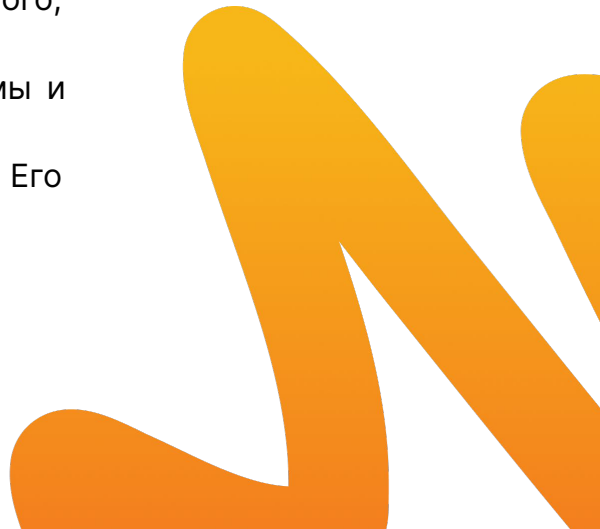
Переменные экземпляра

- Переменные экземпляра являются нестатическими переменными и объявляются в классе вне какого-либо метода, конструктора или блока.
- Поскольку переменные экземпляра объявляются в классе, эти переменные создаются при создании объекта класса и уничтожаются при уничтожении объекта
- В отличие от локальных переменных, мы можем использовать модификаторы доступа для переменных экземпляра. Если мы не укажем модификатор доступа, будет использоваться модификатор доступа по умолчанию.
- Инициализация переменной экземпляра не является обязательной. Его значение по умолчанию равно 0.
- Доступ к переменным экземпляра можно получить только путем создания объектов.



Статические переменные

- Статические переменные также известны как переменные класса.
- Эти переменные объявляются аналогично переменным экземпляра. Разница в том, что статические переменные объявляются с использованием ключевого слова `static` внутри класса вне какого-либо метода, конструктора или блока.
- В отличие от переменных экземпляра, у нас может быть только одна копия статической переменной для каждого класса, независимо от того, сколько объектов мы создаем.
- Статические переменные создаются в начале выполнения программы и автоматически уничтожаются по окончании выполнения.
- Инициализация статической переменной не является обязательной. Его значение по умолчанию равно 0.
- Доступна без создания экземпляра.



Сравнение переменных

Локальные переменные	Переменные экземпляра	Статические переменные
Собственная копия	Собственная копия	Одна копия
Изменения не отражаются	Изменения не отражаются	Изменения отражаются
Доступ через ссылку	Доступ через ссылку	Доступ на прямую

Пример типов переменных

Локальные
переменные:

```
public static void main(String[] args) {  
    int var = 10; // Declared a Local Variable  
    // This variable is local to this main method only  
    System.out.println("Local Variable: " + var);  
}
```

Переменные
экземпляра:

```
String test;  
// Default Constructor  
public Practice() {  
    this.test = "Aleks Reingand"; // initializing Instance Variable  
}
```

Статические
переменные:

```
public class ExampleStatic {  
    static String name = "Thomas";  
}
```

```
public class TestStaticVar {  
    public static void main(String[] args) {  
        String name = ExampleStatic.name;  
        System.out.println(name);  
    }  
}
```


Сканер

- Это класс в пакете `java.util`, используемый для получения ввода типов, таких как `int`, `double` и др., и строк.
- Чтобы создать объект класса `Scanner` → передаем предопределенный объект `System.in`, который представляет собой стандартный входной поток.
- Для чтения числовых значений определенного типа данных ТИП используется функция `nextТИП()`.
- Например, чтобы прочесть значение типа `short`, мы можем использовать `nextShort()`.
- Для чтения строк мы используем `nextLine()`.
- Чтобы прочесть один символ, мы используем `next().charAt(0)`.



Экспресс-опрос

- **Вопрос 1.**

Можете ли вы использовать локальную переменную из метода один в методе два?

- **Вопрос 2.**

Объясните, что такое Переменная экземпляра.



4

ВОПРОСЫ ПО ОСНОВНОМУ БЛОКУ

5

ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

ЗАДАНИЕ

1. Создать объект Scanner
2. Получить от пользователя строку
3. Получить от пользователя число
4. Получить от пользователя булеан значение
5. Получить от пользователя длинное число
6. Получить от пользователя дробное число



TEL-RAN
by Starta Institute

6

ПРАКТИЧЕСКАЯ РАБОТА

Практическое задание 1

1. Создать объект Scanner
2. Получить от пользователя первое число
3. Получить от пользователя второе число
4. Произвести математические операции $+$, $-$, $*$, $/$
5. Вывести результат в консоль



Реализация задания 1

```
Scanner scanner = new Scanner(System.in);

int numberOne = scanner.nextInt();
int numberTwo = scanner.nextInt();

System.out.println(numberOne + numberTwo);
System.out.println(numberOne - numberTwo);
System.out.println(numberOne * numberTwo);
System.out.println(numberOne / numberTwo);
```


7

ОСТАВШИЕСЯ ВОПРОСЫ

Полезные ссылки

- [Class \(computer programming\) - Wikipedia](#)
- [Method \(computer programming\) - Wikipedia](#)

ЗАКЛЮЧЕНИЕ



Дополнительная практика

Объявите 10 переменных типа `int` со значениями 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Объявите переменные `int` и `double` сохраните полученную сумму в эти переменные, разделив результат на 10.

Вы получили среднее арифметическое этих чисел в двух типах `int` и `double`.

Распечатайте через `println`.

Проанализируйте результат.