

Java Basic

lecture #2. Variables. Keyboard input

Mentor: <....>

lecture #2. Variables. Keyboard input

- Numeral systems, data in memory
- Variables
- Naming conventions
- Scanner. Input int values from user

Системы счисления

Система счисления — это система письма для выражения чисел;
то есть математическая запись для представления чисел данного набора с использованием цифр или других символов согласованным образом.

- Позиционные системы
- Не позиционные системы

Позиционные системы счисления

В позиционных системах счисления один и тот же числовой знак (цифра) в записи числа имеет различные значения в зависимости от того места, где он расположен.

Наши 10 пальцев.

2 — двоичная (в дискретной математике, информатике, программировании);

3 — троичная;

8 — восьмеричная;

10 — десятичная (используется повсеместно);

12 — двенадцатеричная (счёт дюжинами);

16 — шестнадцатеричная (используется в программировании, информатике);

20 — двадцатеричная;

60 — шестидесятеричная (единицы измерения времени, измерение углов и, в частности, координат, долготы и широты).

Не позиционные системы счисления

В непозиционных системах счисления величина, которую обозначает цифра, не зависит от положения в числе.

При этом система может накладывать ограничения на положение цифр, например, чтобы они были расположены в порядке убывания.

Римские цифры

Натуральные числа записываются при помощи повторения этих цифр.

При этом, если большая цифра стоит перед меньшей, то они складываются (принцип сложения), если же меньшая стоит перед большей, то меньшая вычитается из большей (принцип вычитания).

$V = 5$; $4 = IV$; $6 = VI$;

Десятичная система - Decimal

Используется 10 цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Число $\rightarrow 333 = 3 \cdot 100 + 3 \cdot 10 + 3 \cdot 1 = 3 \cdot 10^2 + 3 \cdot 10^1 + 3 \cdot 10^0$

$$1 = 10^0$$

$$10 = 10^1$$

$$100 = 10^2$$

$$1000 = 10^3$$

$$10000 = 10^4$$

$$100000 = 10^5$$

$$1000000 = 10^6$$

$$10000000 = 10^7$$

Как записывается

- 123
- 456

Восмеричная система - Octo

Используется 8 цифр: 0, 1, 2, 3, 4, 5, 6, 7

8->10

Число -> $333 = 3 \cdot 8^2 + 3 \cdot 8^1 + 3 \cdot 8^0 = 3 \cdot 64 + 3 \cdot 8 + 3 \cdot 1 = 192 + 24 + 3 = 219$

10->8

$219 = 219 / 64 = 3 \text{ ост} = 219 - 192 = 27 / 8 = 3 \text{ ост} = 27 - 24 = 3 / 1 \text{ ост} = 3 - 3 = 0$

$$1 = 8^0$$

$$8 = 8^1$$

$$64 = 8^2$$

$$512 = 8^3$$

$$4096 = 8^4$$

$$32768 = 8^5$$

$$262144 = 8^6$$

$$2097152 = 8^7$$

Как записывается

- 0123
- 0456

Двоичная система - Binary

Используется 2 цифры: 0, 1

2 -> 10

Число -> 101101 = $1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$
= $32 + 0 + 8 + 4 + 0 + 1 = 45$

$$1 = 2^0$$

$$2 = 2^1$$

$$4 = 2^2$$

$$8 = 2^3$$

$$16 = 2^4$$

$$32 = 2^5$$

$$64 = 2^6$$

$$128 = 2^7$$

$$512 = 2^8$$

$$1024 = 2^9$$

Как
записывается

используя
индекс 0b

Data in memory

Ячейка памяти = есть ток, или нет = 1 или 0

0 или 1 – это и есть наш бит информации

8 бит = 1 байт

Группа из 8 бит выглядит как 00000000, 00000001, ..., 11111111

Соответствуют десятичным = 0, 1, ..., 255 значений.

Long = 8 байт или 64 бит = -2^{63} до $2^{63}-1$

Шестнадцатеричная система - Hex

Используется 16 цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

16 → 10

Число → $333 = 3 \cdot 16^2 + 3 \cdot 16^1 + 3 \cdot 16^0 = ?$

$$1 = 16^0$$

$$16 = 16^1$$

$$256 = 16^2$$

$$4096 = 16^3$$

$$65536 = 16^4$$

$$1048576 = 16^5$$

Как
записывается

используя
индекс 0x

Соглашения об именах в Java

Class: если вы называете какой-либо класс, то это должно быть **существительное**, и поэтому оно должно быть названо в соответствии с целью, которая должна быть достигнута в программе.

```
class Student { } class Scanner { }
```

Interface: если вы называете интерфейс, оно должно выглядеть как **прилагательное**.

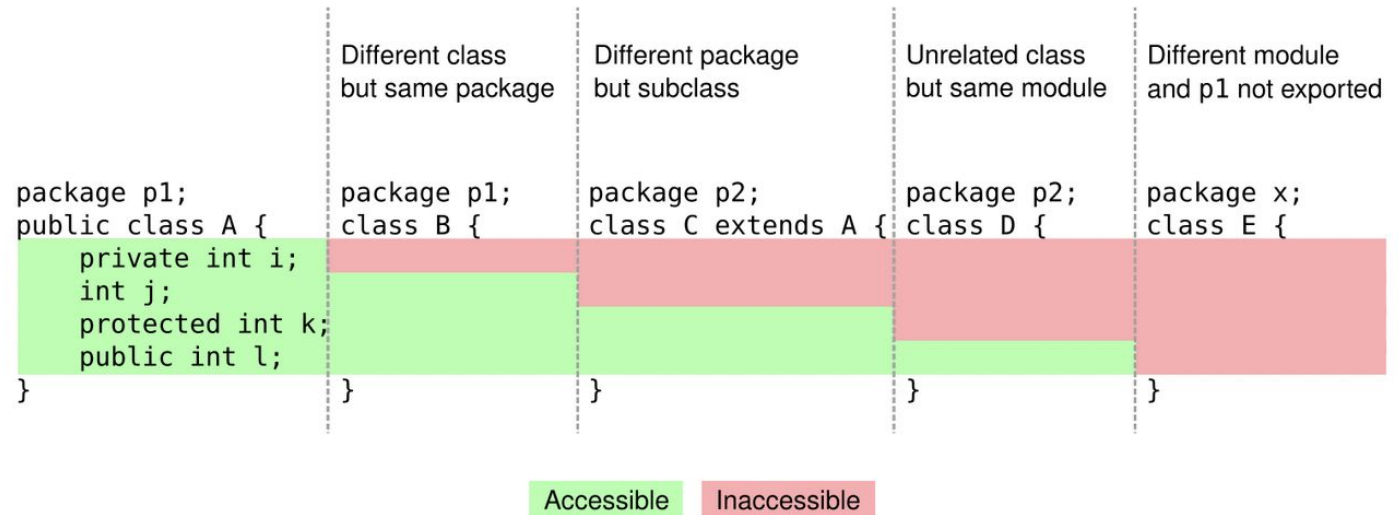
```
interface Runnable() interface Flying()
```

Method: предполагается, что метод должен делать что-то, что он содержит в своем теле, это должен быть **глагол**.

Constant: как следует из названия, это должно выглядеть так, как мы читаем, как будто оно **фиксировано** PI, MAX_INT, MIN_INT.

Модификаторы доступа

Modifier	Class	Package	Subclass	Global
Public	Yes	Yes	Yes	Yes
Protected	Yes	Yes	Yes	No
Default	Yes	Yes	No	No
Private	Yes	No	No	No



Scope of variables

Локальные переменные
Переменные экземпляра
Статические переменные

Локальные переменные

1. Переменная, определенная в блоке, методе или конструкторе, называется локальной переменной.
2. Эти переменные создаются при входе в блок или при вызове функции и уничтожаются после выхода из блока или при возврате вызова из функции.
3. 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.
4. Инициализация локальной переменной обязательна перед ее использованием в определенной области

Scope of variables

Переменные экземпляра

1. Переменные экземпляра являются нестатическими переменными и объявляются в классе вне какого-либо метода, конструктора или блока.
2. Поскольку переменные экземпляра объявляются в классе, эти переменные создаются при создании объекта класса и уничтожаются при уничтожении объекта.
3. 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.
4. Инициализация переменной экземпляра не является обязательной. Его значение по умолчанию равно 0.
5. Доступ к переменным экземпляра можно получить только путем создания объектов.

Scope of variables

Статические переменные

1. Статические переменные также известны как переменные класса.
2. 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.
3. В отличие от переменных экземпляра, у нас может быть только одна копия статической переменной для каждого класса, независимо от того, сколько объектов мы создаем.
4. Статические переменные создаются в начале выполнения программы и автоматически уничтожаются по окончании выполнения.
5. Инициализация статической переменной не является обязательной. Его значение по умолчанию равно 0.
6. Доступна без создания экземпляра

Static variables VS Instance variables

Static variables	Instance variables
Одна копия	Собственная копия
Изменения отражаются	Изменения не отражаются
Доступ на прямую	Доступ через ссылку

Scanner

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)`**.

Полезности

Вики инфо: системы счисления

https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%81%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F

Таблица соответствия
восьмеричных цифр и двоичного
кода

Дв	Вос	Дес
000	0	0
001	1	1
010	2	2
011	3	3
100	4	4
101	5	5
110	6	6
111	7	7

Таблица соответствия
шестнадцатеричных цифр и
двоичного кода

Дв	шес	Дес
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	a	10
1011	b	11
1100	c	12
1101	d	13
1110	e	14
1111	f	15