

1. Специализация: данные и функции

1.1. В предыдущем разделе

- Краткая история (причины возникновения);
- инструментарий, выбор версии;
- CLI;
- структура проекта;
- документирование;
- некоторые интересные способы сборки проектов.

1.2. В этом разделе

Будет рассмотрен базовый функционал языка, то есть основная встроенная функциональность, такая как математические операторы, условия, циклы, бинарные операторы. Далее способы хранения и представления данных в Java, и в конце способы манипуляции данными, то есть функции (в терминах языка называющиеся методами).

1.3. Данные

1.3.1. Типы, преобразование типов

Хранение данных в Java осуществляется привычным для программиста образом: в переменных и константах. Языки программирования бывают типизированными и нетипизированными (бестиповыми).

Отсутствие типизации в основном присуще старым и низкоуровневым языкам программирования, например, Forth, некоторые ассемблеры. Все данные в таких языках считаются цепочками бит произвольной длины и, как следует из названия, не делятся на типы. Работа с ними часто труднее, и при чтении кода не всегда ясно, о каком типе переменной идет речь. При этом часто безтиповые языки работают быстрее типизированных, но описывать с их помощью большие проекты со сложными взаимосвязями довольно утомительно.



Java является языком со **строгой** (также можно встретить термин «**сильной**») **явной статической** типизацией.

Что это значит?

- Статическая - у каждой переменной должен быть тип и мы этот тип поменять не можем. Этому свойству противопоставляется динамическая типизация;
- Явная - при создании переменной мы должны ей обязательно присвоить какой-то тип, явно написав это в коде. Бывают языки с неявной типизацией, например, Python;
- Строгая(сильная) - невозможно смешивать разнотипные данные. С другой стороны, существует JavaScript, в котором запись `2 + true` выдаст результат 3.

Все данные в Java делятся на две основные категории: примитивные и ссылочные.

Данные: типы, преобразование типов, константы и переменные (примитивные, ссылочные), бинарное представление, массивы (ссылочная природа массивов, индексация, манипуляция данными);

Тип	Пояснение	Диапазон
byte	Самый маленький из адресуемых типов, 8 бит, знаковый	[-128, +127]
short	Тип короткого целого числа, 16 бит, знаковый	[-32 768, +32 767]
char	Целочисленный тип для хранения символов в кодировке UTF-8, 16 бит, беззнаковый	[0, +65 535]
int	Основной тип целого числа, 32 бита, знаковый	[-2 147 483 648, +2 147 483 647]
long	Тип длинного целого числа, 64 бита, знаковый	[-9 223 372 036 854 775 808, +9 223 372 036 854 775 807]
float	Тип вещественного числа с плавающей запятой (одинарной точности, 32 бита)	
double	Тип вещественного числа с плавающей запятой (двойной точности, 64 бита)	
boolean	Логический тип данных	true, false

Рис. 1: Основные типы данных в языке Java

Базовые функции языка: математические операторы, условия, циклы, бинарные операторы;

Функции: параметры, возвращаемые значения, перегрузка функций;

1.3.2. Антипаттерн "магические числа"

В прошлом примере мы использовали антипаттерн - плохой стиль для написания кода. Число 18 используется в коде без пояснений. Такой антипаттерн называется "магическое число". Рекомендуется помещать числа в константы, которые хранятся в начале файла. `ADULT = 18` `age = float(input("Ваш возраст: "))` `how_old = age - ADULT` `print(how_old, "лет назад ты стал совершеннолетним")`

Плюсом такого подхода является возможность легко корректировать большие проекты. Представьте, что в вашем коде несколько тысяч строк, а число 18 использовалось несколько десятков раз. □ При развертывании проекта в стране, где совершеннолетием считается 21 год вы будете перечитывать весь код в поисках магических "18" и править их на "21". В случае с константой изменить число нужно в одном месте. □ Дополнительной сложности могут возникнуть, если в коде будет 18 как возраст совершеннолетия и 18 как коэффициент для расчёта чего-либо. Теперь править кода ещё сложнее, ведь возраст изменился, а коэффициент - нет. В случае с сохранением значений в константы мы снова меняем число в одном месте.

1.4. Базовый функционал языка

1.4.1. Математические операторы

1.4.2. Условия

1.4.3. Циклы

1.4.4. Бинарные арифметические операторы

;

1.5. Функции

параметры, возвращаемые значения, перегрузка функций

Задания к семинару

- Написать как можно больше вариантов функции инвертирования массива единиц и нулей за 15 минут (без ветвлений любого рода);
- Сравнить без условий две даты, представленные в виде трёх чисел гггг-мм-дд;