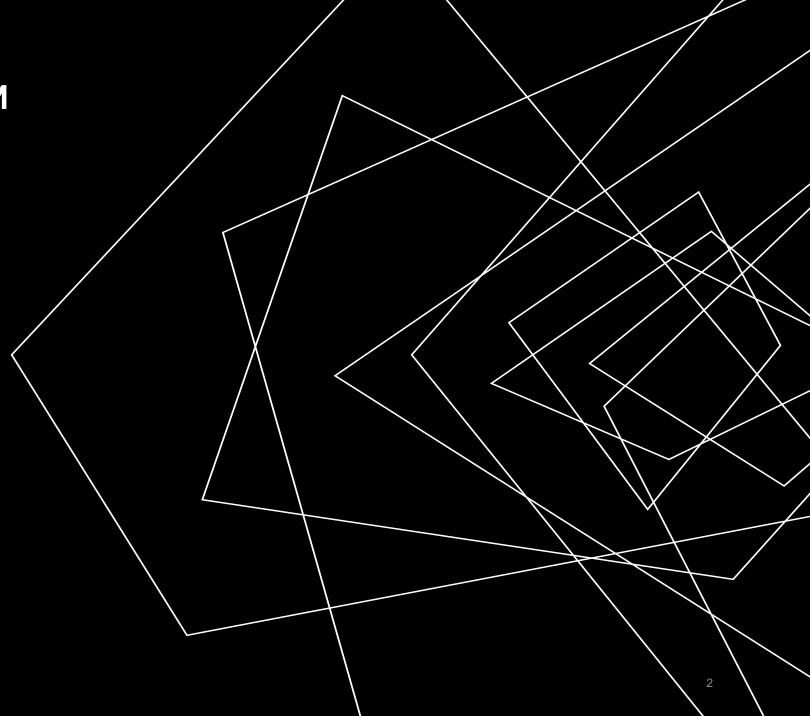




- Типы данных в Lisp
- Операции над типами
- Преобразование/приведение типов
- Сравнения типов
- Список литературы



ТИПЫ ДАННЫХ: ПРИМИТИВНЫЕ

- Числа (Number):
 - o integer целые числа
 - float числа с плавающей точкой
 - ratio рациональные дроби
- **Символ (Symbol)** уникальное имя, часто используемое как идентификатор переменной или функции
- Строка (String) последовательность символов в двойных кавычках
- Булевы значения (Boolean):
 - o t − true
 - o **nil** false
- Nil пустой список ()
- Cons-ячейка (Cons Cell) фундаментальная строительная единица списков. Представляет собой пару из двух указателей: *car* и *cdr*

ТИПЫ ДАННЫХ: ПОЛЬЗОВАТЕЛЬСКИЕ

1. Структуры (Struct) – создаются макросом defstruct

```
44 ;; Структуры (Struct)
45 (defstruct person
46 name
47 age)
48
49 (defvar ivan (make-person :name "Ivan" :age 30)) ; Создание экземпляра структуры
50
51 (print (person-name ivan)) ; Вывод: "Ivan"
52 (print (person-age ivan)) ; Вывод: 30
```

2. Классы (CLOS) — создаются макросом defclass

```
;; Классы (CLOS)
(defclass animal ()
((species :initarg :species :accessor species)
(sound :initarg :sound :accessor sound)))

(defvar cow (make-instance 'animal :species "Cow" :sound "Muuu"))

(print (species cow)) ; Вывод: "Cow"
(print (sound cow)) ; Вывод: "Миии"
```

ОПЕРАЦИИ НАД ТИПАМИ

- (type-of object)
 - о Возвращает тип переданного объекта
 - Пример: (type-of 5) -> integer
- (typep object type)
 - о Проверяет, принадлежит ли объект указанному типу
 - Возвращает *t* или *nil*
 - Пример: (typep 100 'integer) -> t

ПРЕОБРАЗОВАНИЕ ТИПОВ

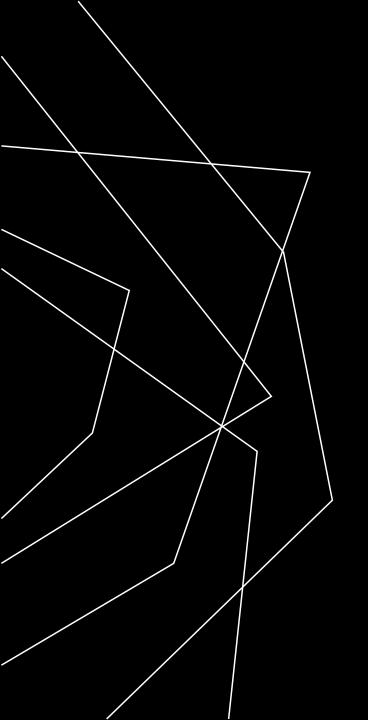
- (coerce object result-type)
 - о Пытается преобразовать объект к указанному типу, если это осмысленно
 - Пример: (coerce 3.8 'integer) -> 3
- (parse-integer string)
 - о Функция для преобразования строки в целое число
 - Пример: (parse-integer "52") -> 52
- (format nil "~A" object)
 - о Универсальный способ преобразовать что угодно в строку
 - Пример: (format nil "~A" 52) -> "52"
- Другие функции:
 - o (float 5) -> 5.0
 - (character "A") -> #\A

СРАВНЕНИЕ ЗНАЧЕНИЙ И ТИПОВ

- (eq x y)
 - о Проверяет, являются ли **х** и **у** одним и тем же объектом в памяти
- (eql x y)
 - Проверяет, являются ли либо х и у одинаковыми числами/символами, либо одним и тем же объектом
 - о Пример: *(eql 5 5) -> t*
- (equal x y)
 - о Сравнивает структуры объектов
 - о Рекурсивно устанавливает поэлементное равенство для списков и строк
 - о Пример: *(equal (list 1 2) (list 1 2)) -> t*
- (equalp x y)
 - Более мягкая версия *equal*
 - Игнорирует регистр символов в строках и разницу между типами чисел.
 - Пример: (equalp 5 5.0) -> t

СПИСОК ЛИТЕРАТУРЫ

- LISP Краткое руководство
- Основы программирования на языке Lisp



СПАСИБО ЗА ВНИМАНИЕ