classmate
Date 13·8·2019
Page 1

BOTTOM-UP STEPS

LITERALS

a language focused on literals diverse, and all else just imported from C. that help into good depth and that is interesting to many persons.

literals - also known as primitives - or if you coant to take +2 words, it will be "primitive data types" — no actually "primitive data values".

bitfields
unions

we all lobe bits, ones and zeros.
↳ 1 0

and there is something called a nibble (1100).

and then bytes and words and dwords....

↳ come with the BINARY REPRESENTATION.

1100b
↳ binary representation (of size? (datatype))

So 2 things — representation & type are 2 separate things. - late discovery. - better late than never.

PRIMITIVE = REPRESENTATION + TYPE. (DATA).

lets consider some reperesentations rinku has read from various places (URLs) over the internet:

- binary - 10001b 010b
- octal - 4360
- hexadecimal - 1abch
- other bases. .... $(0123)_4$
- roman numerals :- $(VII)_R$
- decimal! - 113_{10}
- ascii - g- ascii
- base64 - $(hel-)_{b64}$

convenient readability separators

- 1 000 234 . 00 → thinking of USD :/ $

who decides how a primitive is parsed.
how may such parsing be modified at compiletime.
how can literals be defined.

defining other literals for boolean variable types

yes right correct ...

no wrong incorrect

defining floating point literals

not only f32 f64 but other sizes.

defining fixed point literals.

defining bigint / big number literals.

defining character / string literals.

defining literals with

- units
- errors
- ranges
- angles
- complex numbers.
- other associated meta information

is this all going to be syntactic sugar for a constructor call?

classmate
Date
Page

other more complex literals →

- lists
- maps
- tuples
- matrices
- images
- trees
- stacks
- queues
- graphs
- stock symbols
- graphs
- formatted text
- html
- JSON
- JSX
- xml
- css
- binary blob (right in src code)
- svg
- wav/midi
- urls

what to do @ compile time, what to do @ runtime?

execution pass 1, execution pass 2, ...

age: 24 years

height: 32m

date: 1.2.2019 date

date: Date (1, 2, 2019)

age: Duration (years = 24)

print (age in months)

print months (age) → 24×12 = 48 = 288 months.

age: ~ 24 years

print months (age) → a separate associated fn. ~ 288 months

variable associations (structs?)

use simpler translation, not fun closure one.

also run @ compile time, not runtime?

length: 32 m

height: 7 12 m

area: length x height

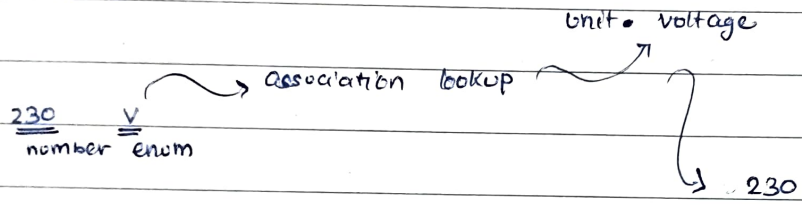
print area ——> 32x7/2 m^2 unicode

voltage: 230 V

current: 4 A

print (voltage * current

// 920W



unit. voltage

↓

unit. voltage voltage = unit. voltage (230, unit. voltage. V),

unit. current current = unit. current (4.0, unit. current. A),

unit. power y = voltage * current ( )

unit