Types

data type: size, structure, encoding, semantics

type specification: Ts := (T,Is,F) T: base set,
Is: invarians, Is(ts) == true
F: operations F = { F1, F2, ... Fn }

Eg: type V is a Stack type where E is the type of the elements.?

Possible operations: (without "full")?

empty: ->V (construction of the empty stack)?
isempty: V -> L (is stack empty?)?
push: V x E -> V (put an element into stack)?
pop: V -> V x E (remove an element from stack)?
top: V -> E (ask the last element)?

Constraints: ?
Domain of pop and top: V \ {empty}

Axiomatic specification:

isempty(empty) or v=empty -> isempty(v)
isempty(v) -> v = empty
not isempty(push(v, e))
pop(push(v,e)) = (v, e)
push(pop(v)) = v
top(push(v,e)) = e

type relaization: TT := (ro,I,S) ro: E*xTs representation function
I: E->Bool invariant
S: S = {S1,...,Sn} operations
statically/dynamically typed
strong/weak typesystem

eg. Python: dynamically typed strong(ly) type(d)system
~ polymorfism: not very statically typed

Strong typing vs. type alias
typedef/using C++
typedef double length;
using length = double;

Type equivalence nominal or structural
ADA: R1: Array (1..10) of Integer
R2,R3: Array (1..10) of Integer R2<=>R3

C: strict aliasing

Subtype/derived type in ADA:
```
  subtype Days is Integer range 1..31; //not new type just a restriction
  type Speed is Float;
  type Distance is new Float;   // new type
```

type conversions : widening or narrowing. int i{3.14}, // error: narrowing
OOP: upcast OK,
    downcast: ? (checked downcast: dynamic_cast, or (CC) in Java)

type taxonomy: (there are also different taxonomies)
```
 Primitive
   Pointer
   Scalar
     Real  (fixed and floating point representation)
       floating point: IEEE 754 https://en.wikipedia.org/wiki/IEEE_754
     Discret/Integral (incl. char, bool, signed, enum, ...)
 Composit/Aggregate
   (Enumeration)
   Iterated
     Array
   Union
     tagged (Algol68, Pascal variant record, C++ variant)
     non tagged (C union)
   cartesian product types, Tupples
     Struct/Class
   ...
   Set  (Pascal,ADA)

 String
    Built-in? Pascal/python
    Library:  C++, Java
    Nothing:  C
  Representation:
     C-like, terminator char: '\0'                    abc\0
     Pascal-like: starts with size (but 1byte in Pascal)  3abc
     User defined class (C++, Java)
```

Value semantics vs. reference semantics
```
 Java: primitive types vs reference types
 C#:   value types vs reference types
```

Standardization:
```
 Java  short 2 bytes, int 4 bytes
 C/C++ 1 == sizeof(char) < sizeof(short) <= sizeof(int) <= ...
 C++11 int16_t, int32_t, int64_t,  ...
```

Signed/unsigned (not in java)