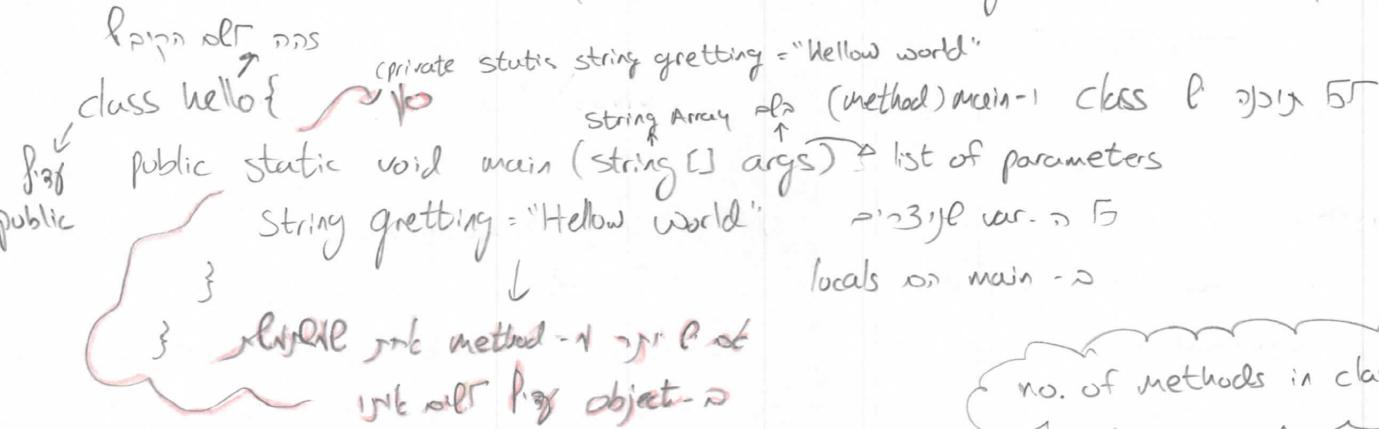


Chapitre 1

x.java → x.class
Netbeans: → Compiler:

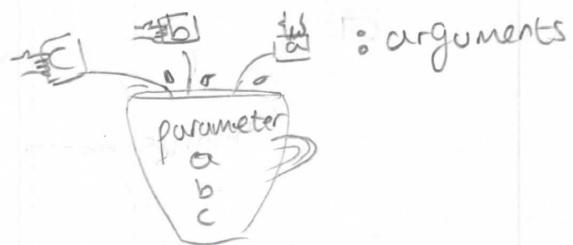
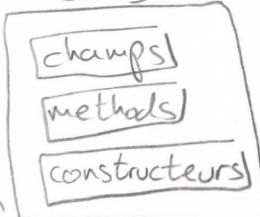
SourceCode → Bytecode → Langue Machine (Mac / Windows)



Anatomy of class members

	modificateur	static	type	name	value
champ	private	static	String	Greeting	Hello world
method	public	static	void	main	{ } -> block
proto	protected	C# B block		(String [] args)	- list of parameters: -> for method -> - block

class → מושך מושך מושך



class hello {

 champ ← private String greetings = "Hello world";
 constructeur ← public hello () {

 System.out.println(greetings);
 }

 method ← public static void main (String [] args) {
 new hello ();
 }
 }

String C# Bile sm3 java 1
args if args greeting null

if args not bind Bile 2
main prouf Jelma link as

constat hello -> big main

Static or nothing?

static and method the chaff of null void for static method

① public static void main(...){
...println(greeting)}

No ref on greeting given pr-ref for

② public static void main(...)
new Hello();

ref for the creation of ref for it
Go to the code

: if it's 3rd for method its obj constructor

method main (void)

Cons (pic)

type if ref #

method main (main)

Cons(Hello)

class -> alt class ref #

(Cons[new hello();])

new obj & ref len #

parameter = method (String n)

argument = method (String n)

if return

!x != 0.00

1 value / array

Source code → ?

Compilation

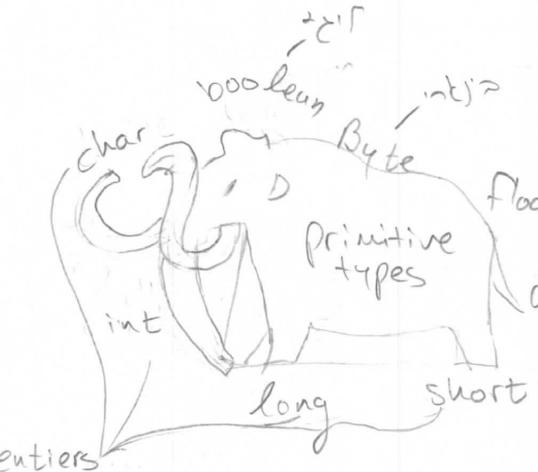
; jml, Syntax JMLC

execution (runtime error)



logique

logique/pratique ref len #



all types are classes

Array

Ref types

class

Interface

primitive ← object
primitive name + name = new primitive... (value);
in capital
Integer x = new Integer (24);

(An instance) is a <class>

a dog is an animal

parent class

child class



Object O.name;
new Object()

constructor

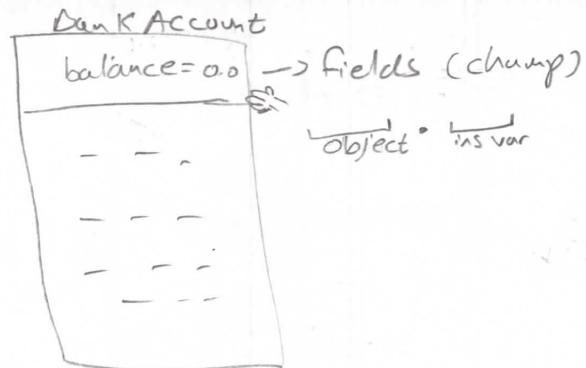
class -> ref

ref, address

Method +
class
do another
thing
do
1 thing

Object O.name;
new Object()
ref = null
ref =

① Encapsulation



constructor method

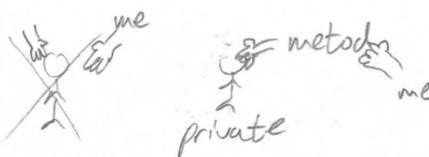
function obj Bp 3n
for 231

[5] [2] [3]
this var this var this var

Shared (static)



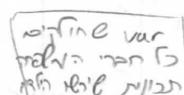
OO



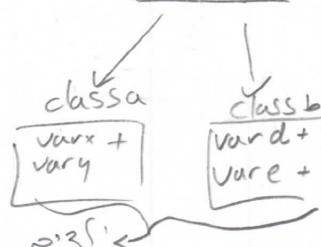
All objects are mutable?

② inheritance

Class

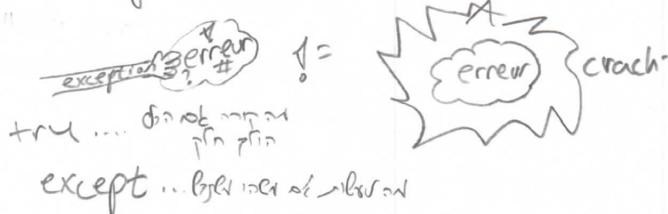


min (parent, base, superclass)

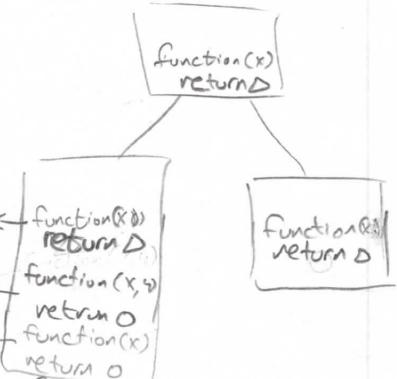


(child, derived, subclass)

③ exception



④ polymorphism



(returning different types) return to which function we write: not poly ←
return to which slot to return (ref)

• What are the units of description are (code, object, class)

• Units = classes

• Wrap conceptual ideas in one class → things

• What operation can be done → actions

Java

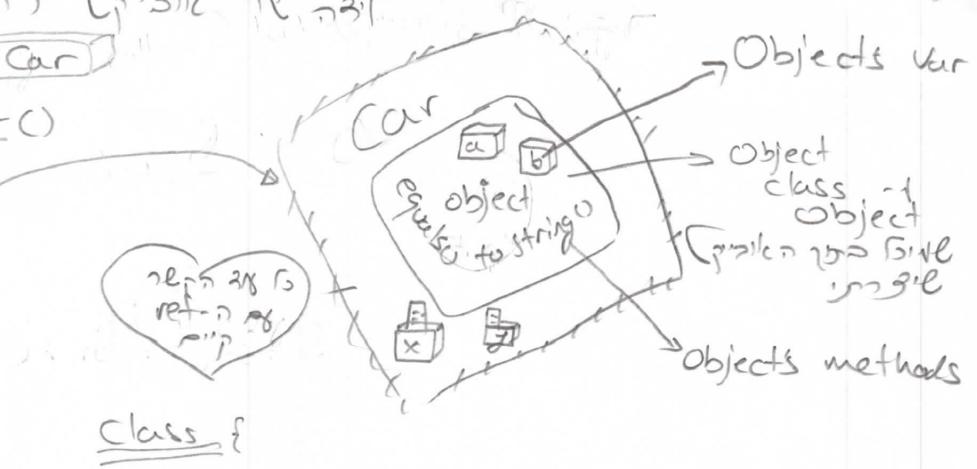
1. גדרת רכבי (Car) כרך רכבי

① Declaration: Car myCar

② Creation: new Car()

Constructor -> בירך גודל גודל גודל

③ Link: = myCar



method() {



Class {



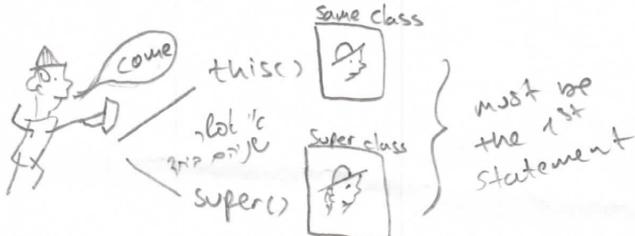
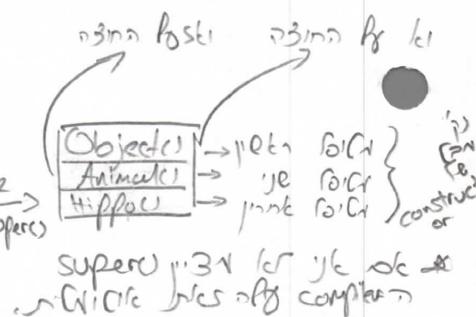
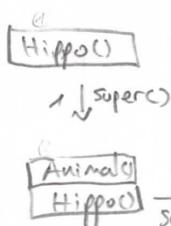
creates the object and can add some useful getters

i++ כרך רכמי בד : מילוי כרך
new br-fetar ורכס ורכס Objects State
has-a כוכב כרך רכמי³
ArrayList -> כרך רכמי

✓ public car (int size) } not allowed. for the compiler it's the same constructor
✗ public car (int volfuel) } it looks on the TYPE and the ORDER
To COMPILE it has to have a different argument list



Hippo כרך גודל גודל
Object יונת גודל גודל
Object 1 Animal
Constructor chaining



must be the 1st statement

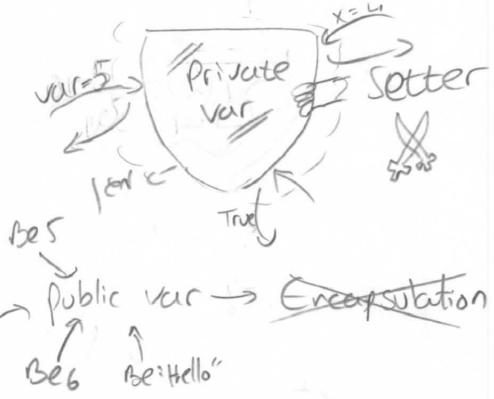
get rid of its ref

ad ref (method:)
method: Life z = new Life()

assignment
Life z = new Life()
z = new Hippo()

ref = null;

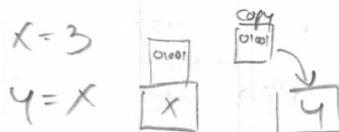
Encapsulation



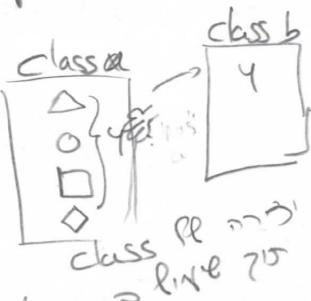
encap = false



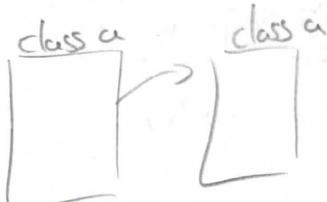
\bullet == ref / object
 val == value / primitive
`valx.equals(valy)` / object



Composition ("has a...")



recursive \rightarrow self



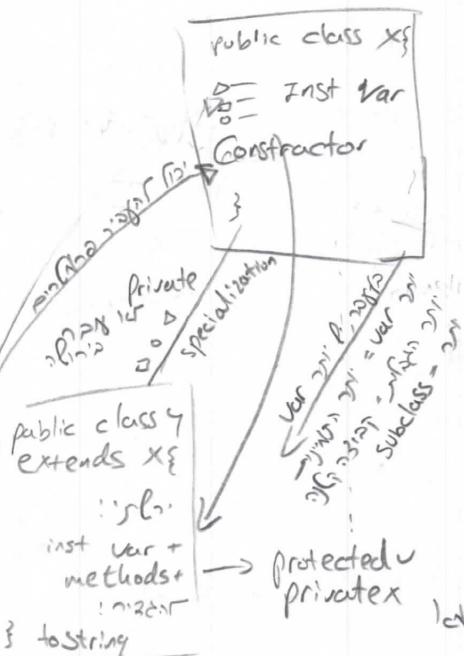
\bullet Super
 \bullet Super var
 \bullet super(var1, var2...);
 parent class -> b64b as jrc

\bullet Constrains
 class -> ol jrc kly
 olc olc l'FB subclass Rg fyoF Rjw kf kln

? : ? # ?
Anonymous

`obj x = new Obj (new Object(...), ...)`

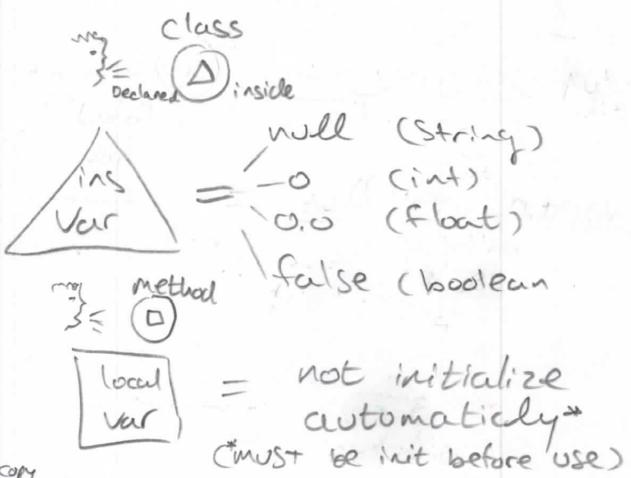
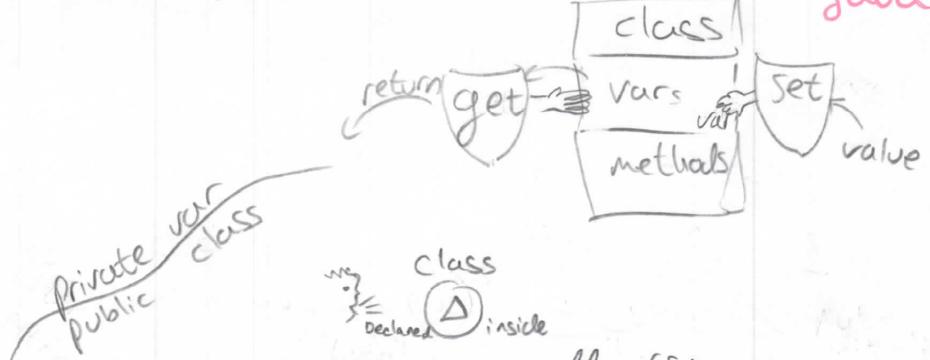
Inheritance ("is a...")



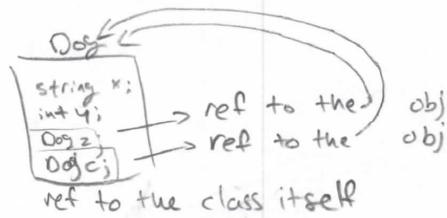
\rightarrow Super's ol jrc
 Super's ol jrc
 Default Const -> SubConst ->
 ... (... jrc jrc jrc) b64b as ol
 Compile -> ol B1 lrc as

\downarrow C5
 SubConst -> Super -> ol B1

Java



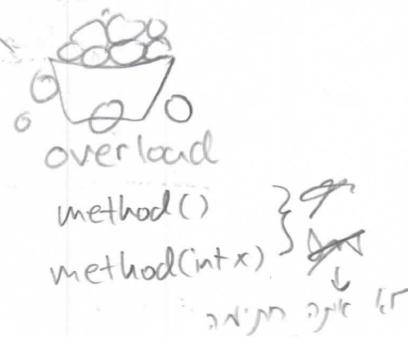
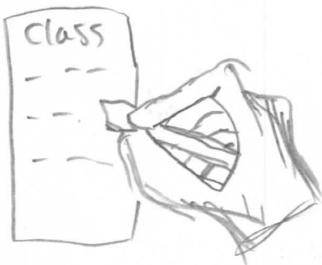
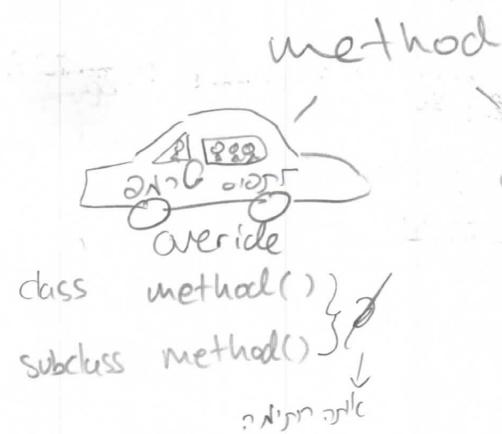
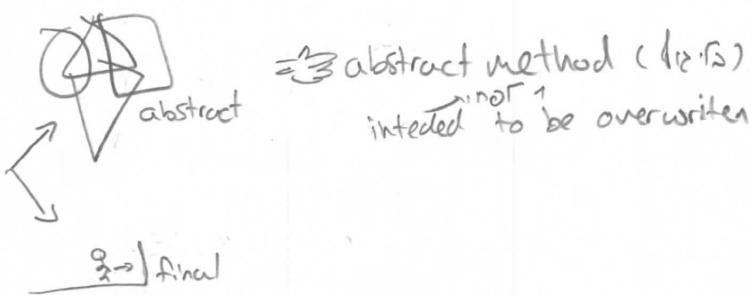
Recursive \rightarrow self



method of the parent class



x.equals(y) → same object (state like apple in Fruit , apple unlike x.y : apple result in box)
String class override it string 1.equals(S2)- apple equals apple not box so \leftarrow true



modifiers

public

protected

default

private

terminology

Statement ; ; ; ; ; ; ; ;

block { } statements

identifier:

word refer something in the programme

Keywords → there are so

- ① What it suppose to do
- ② List inst + methods
- ③ Write methods' precode (Eng, What to do)
- ④ Write test code (Java, How to do)
- ⑤ Implement the class
- ⑥ Test the methods
- ⑦ Debug and reimplement (if any)

prep code	test code	real code
-----------	-----------	-----------

Chaine -> code compiler
non pas logique

print → address ?
I am an hammer

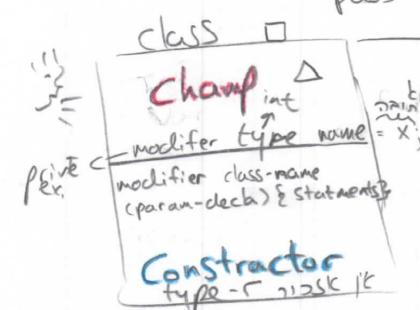
class B → public String toString()
return "I am" + var.

→ print obj Class → PB method →

this.var = var constructor
Parameter de constructor

?
var of field
(insta var)

import Folder name.class
minimal class -> nels
↳ import



Public modifiers, class
abstract declaration
class name
extends Object
associations

Tools A = new Tools(....)

pass au constructeur

constante {
public
protected
private}

Modi
name == class
type "SNT" /c
new N 31265 koy
Constr 285 1677715



Constante /c
n°1 compiler ->
(Δ=defaults) 3rd
value

stop 6.8

modificateurs

final

static

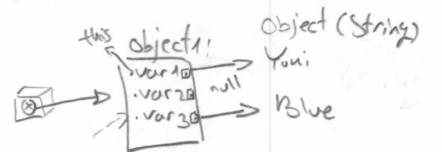
final class
initialise
NBT

method de class
class.method
object.method
argument implicite

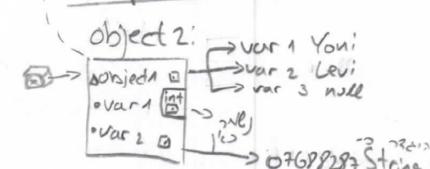
class Object = new Object



de copie
public class Object type



Composition



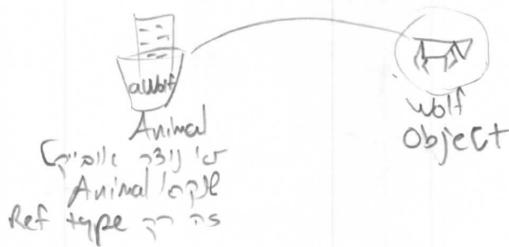
Var local = declare comme une variable locale à method
Champ = var qui est déclarée comme un membre d'une class
modificateur =

Java

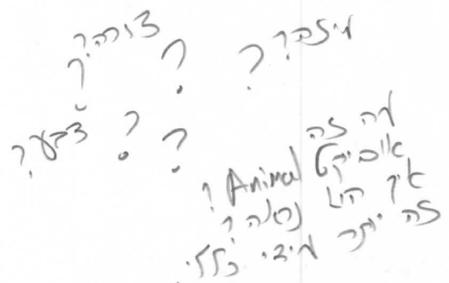
① Wolf aWolf = new Wolf();



② Animal aWolf = new Wolf();



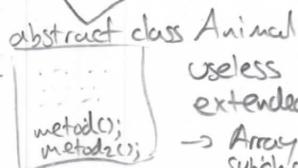
③ Animal aWolf = new Animal();



↓ Abstract ↗ Animal

can't be
in

= new Animal();



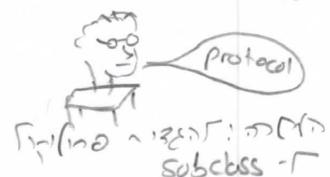
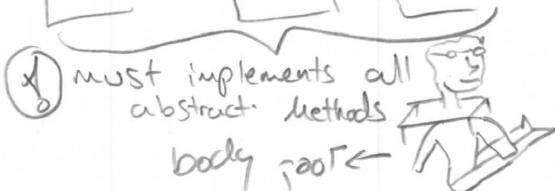
useless ↗ face
extended ↗ ok
→ Array ↗ list ↗ fig.
subclass ↗ can't be first

abstract method class → ↗
abstract not own class → ↗

abstract class Canine

→ methods: must be overridden
→ At least 1 abstract method
(no body)

concrete
class



M M
class Object

superclass →
class → is ref
/ is kind of
polymorphism
String.toString()

IS A...
Class of
Object

All my classe extend Object
→ 'is kind of' like class
class Object is methods → is
refname.toString() → address word

Object x = new Dog();

→ Right bit jlc. ac to
Object class → method →
Dog is methods → is

↓
RULE

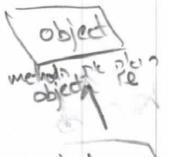
Animal dog = new Dog();
dog.bark(); → Not legal
method → is kind of Animal → is
(Class Dog → o.300 ns) bark()

object dog = new dog();

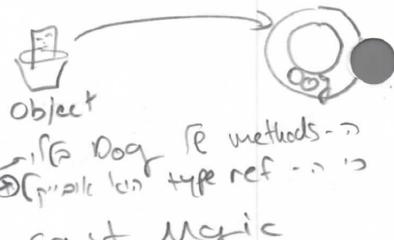
object.bark();

Objects don't bark!!

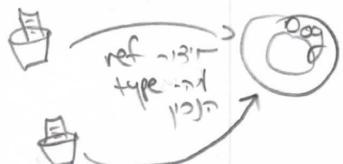
Method is not
reduced to
method



Cast:



cast magic



method -> kind of ref type -> compiler
object type -> is kind of ref type -> pro to