Method:

contains group of instructions.

(invoke or call it) 🡪 some action is taken, depending on programming specified by it

Algorithm:

Recipe for doing something. Whereas program is syntax valid.

Encapsulation

Required for running a set of instructions. This is what method does. And **starting point too**

Public void main( …….

Where is the existing version or syntax created? Like **assembly output** fromwhich we can take existing basic

(pre-defined instructions) and modify / fine tuning to our own version required by **extending**.

Public class {………

Loop are like snooze , where you want to tell something to be done how many times.

Turn right **–** loop **for 3 times** turn left.

**while** **condition** is true loop **until** condition is true

**check** a condition and **do a thing**

**write program for people to read and computer to understand**

**write program for specification. GENERALISE**

Of by one bug

you forgot to do one more thing even though logically it was needed to do.

Comment

Just for the person to read and doesn’t affect the program.

Ex // or /\*\*\*\* ………

/\*\* PreCondition

\*\*PostCondition

Decomposition:

Get to a level of details called **primitives**

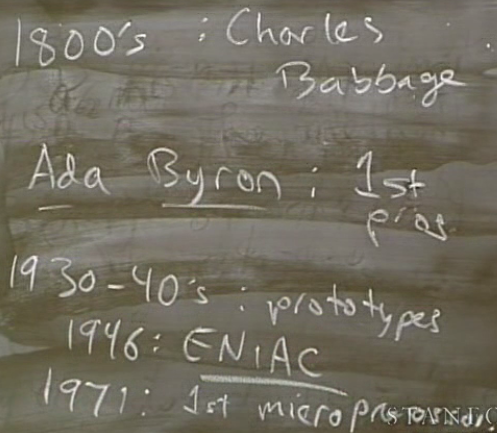
Step wise refinement

2 types top down design and bottom up design

*How much?*

Solve one problem. Methods are 15 lines. Specific/descriptive good name. Comment.

**Think and write like top down design approach**



Computer science is study of problem solving with computational device/methods. Approaches etc

*Programming* is artifact. Something we do to realize the problem solving/ process.

**Machine language** 🡪 micro processor understands. (boils to binarys)

High level language (human readable).

Take instructions from HL and converts to ML is compilation (compiler)

Source code to object code by compiler then to executable file by linker – some other HL langs

But in java, source code to class file (java byte code) its intermediate file then the linker outputs is jar which is passed on to JVM. Runned of computer.

Intermediate language is common irrespective of platform. But jvm for different platforms differs

Java program

Is object oriented programming

Set of class files. A class is a encapsulation of behavior(what manipulation we want to do with data) and data.

Classes get organized into hierarchy.

[Sub class]. super-karel.java (more cooler) 🡪 [Super class(top)] karel.java

Instance

of class means object. It is particular example of the class/ template.

Variable

It’s a box where we stick stuff. Kind of like box where its value changes. Hence so called

Associated with it are, name. type (what kind does it stores). And value (content)

Name space: starts with letter/underscore. Then numbers, letters, \_, non java keywords

Primitives

Built in are int, short, long, float, double, char, Boolean

Double stands for double precision real number.

Declaration of variable without initial value can be done, but before using it initialize it

Declaration: type name = value; (assign operator)

Variables = expression; assignment statement

Expression consists of operators, variables called operands and values

If one of the variable is double it does real value division else integer division

Cast

Doesn’t change intrinsic who you are but changes appearance only.

Specify the type we cast for in front of the variable. Double y = (double) x / 2;

So here treat the whole right side to be type specified inside parentheses and then evaluate

Casting is needed when losing data. Explicitly needs to be told. Int y =(int) x which is double

Explicitly need not be told when not losing data double x = y where y is int

Operators

+, -, \*, /, % (rule of precedence) left to right with multiplication, division more

++, --, +=, -=, \*=, /=

>, ==, != (Boolean expression) or relational operators. ! not, && end, || or

Constant Naming all uppercase. Static means its for class not for instance (one for all. Everything shares)

**Private static final double PI = 3.14d;**

Constructor

Like a factory, create a new object with specified properties

Blocks { ……….. scope – life time of the variables. It will vanish when the closing brace of the place where it was declared is executed.

Graphics – Collage modal

Starts off with blank canvas. We can create various shapes and put up there.

GLabel, GRect, GOval and GLine 🡪 these are classes, templates

x🡪 +ve x axis and y is –ve axis (origin)

everything in graphic window is measured using pixel.

Import keyword, imports a package at that level, not everything inside till root node, package.

Sentinel – its kind of last thing I will give, before I will tell I am done. Its last.

Loop and a half – while(true){….. if(true) break;

For loop used when we know how many iteration is too be done whereas while loop is used when indefinite iteration.

Method: Sine function

One bit of additional complexity is parameters / arguments.

Y = Sine(X) in java Math.sqrt(X), Math.pow(x,y)

Critical part: information hiding. What goes in and what comes out by generalizing the function

Method calling / invoke return variable = Receiver. MethodName(parameter);

**Visibility type name ( parameters) {……..**

**BODY………**

Type – of return value. Void is special type means nothing is being returned

1. encapsulation

achieved through methods (invoked to do an action).

2. our program extends basic programs to achieve

inheritance (previously defined).

3. when we want to execute a known number of times, use for loop

but when dont know about the number of times we execute a step, use while loop.

4. common error:

syntax is correct, but what program was intended to do, didnt happen

5. OfByOneBug:

you have done n times but 1 more time, we forgot to do.

6. comment, for reading purpose only, no interference 0n execution.

File: Program.java

-------------------------------

program runs a chase

/\*\* Method: ascendHurdle

\*\* Pre-condition: Facing east

\*\* Post-condition: Facing east again

\*\*/

7. Step-wise refinement: break into steps until its understandable.

else break into smaller pieces. this Top down design approach

(think abstract and break down)

but bottom up design approach is what every body starts with.

as smaller pieces is easier to do. and then do the next big step

8. algorithm: an approach to some thing using top down approach

move();

doubleBeeper(); -- LOOP[pickBeeper();move();putBeeper();putBeeper();moveBackward();]

moveBackward(); -- turnAround();move();turnaround();

9. good software engineering

10 times more effort to fix a software after initial development

and 10 times many a update happens then the initial release

10. decomposition

solve one problem. 15 avg lines long method. good naming conventions. comments.

11. 1946: ENIAC electronic numeric integrator and calculator

12. 1971: intel 4004, micro processor

13. Comp Science is study of problem solving with computational method

Programming is an artifact.

14. machine lang (defined by hardware) what computer understands, binary

high level language. ex java, c++, basic

translation from HL to ML is compilation (compiler ex eclipse)

15. Normally = Source Code -(compiler)- Object Code -(linker with other obj file)- executable file

object code here is different for different platform ex MAC, PC

16. In Java = Source code -(compiler)- numeric/intermediate lang(class file)

-(linker)- JAR -(JVM understands)- output

but here java object code is same across platform,

but JVM for MAC, PC are different but understands common object code.

17. JAVA: object oriented programming. Class (TEMPLATE/BLUE PRINT)

class - encapsulation of some behaviors(methods N action) and data(fields N variables)

classes are organized with hierarchy.

public class superkarel extends karel { ..............

SK extends K functionality. SK is subclass of K. SK does all K things, and does more.

ex humans extends primates. primates extends animals.

monkey extends primates.

animals[food,blastula embryo]; primates[thumbs,5 fingers]; human[adv brain];

keerthi, object is an example, instance of a class (all characteristics of classes up)

18. variables, box where we stick stuff.

name, type(what kind is stored) and value

primitives: int, long, float, double, boolean, char

give variable a initial value when it makes sense

type variable = value; (statement and expressions and assignment)

19. operators + - \* / %

20. cast

double d =(double)5/2;

21. static, every object instance share the same thing