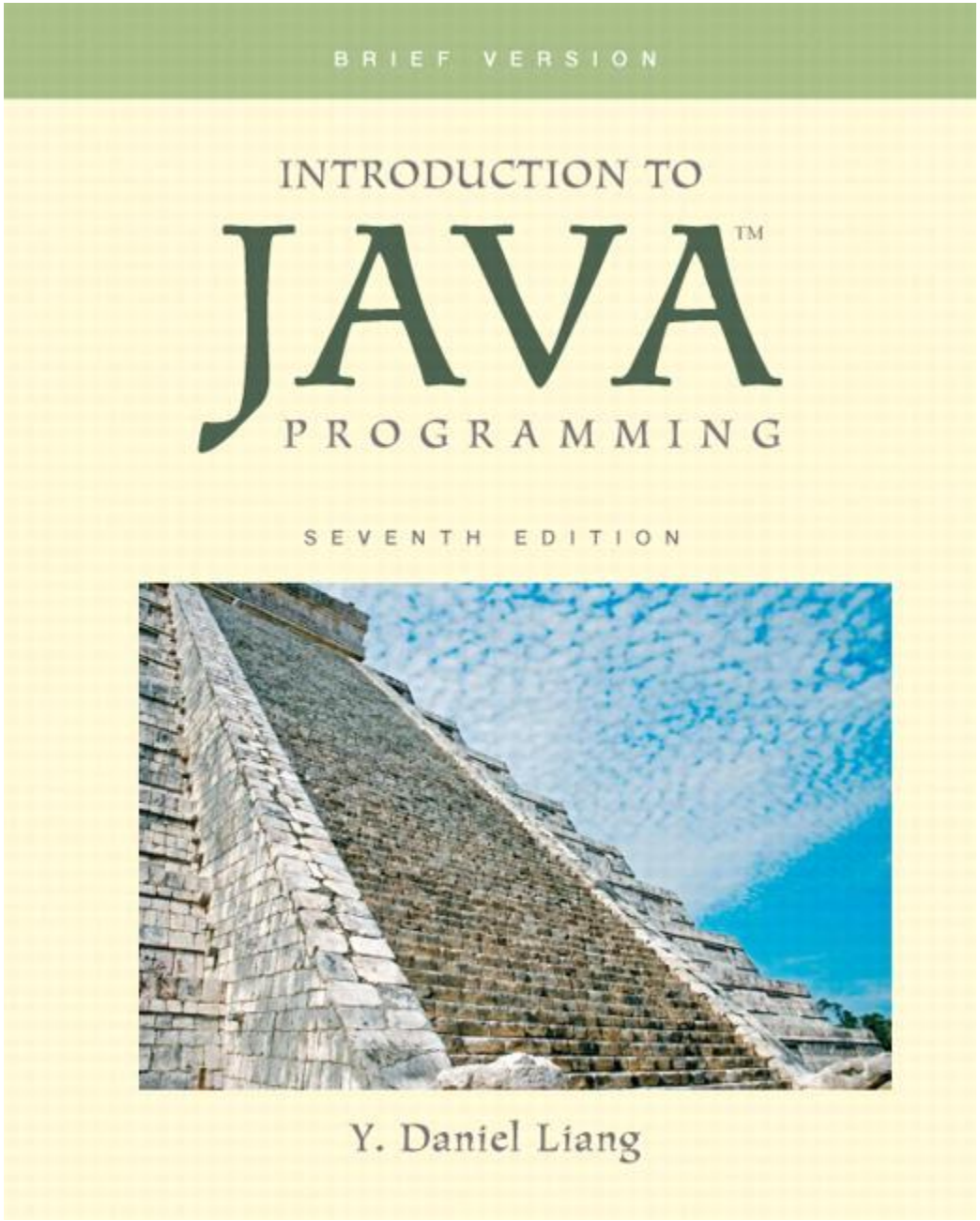
Software Development Programming

Literatura

Introduction to Java Programmng

Y. Daniel Liang

Astrit@rrota.com



SD/Programming

- Çka bën një kompjuter
  - Merr të hyrat (Çka janë?)
    - Të dhëna (Data)
  - I përpunon të dhënat
  - Të dalat (Outputs)
    - I ruan të dhënat
    - I tregon të dhënat

SD/Programming

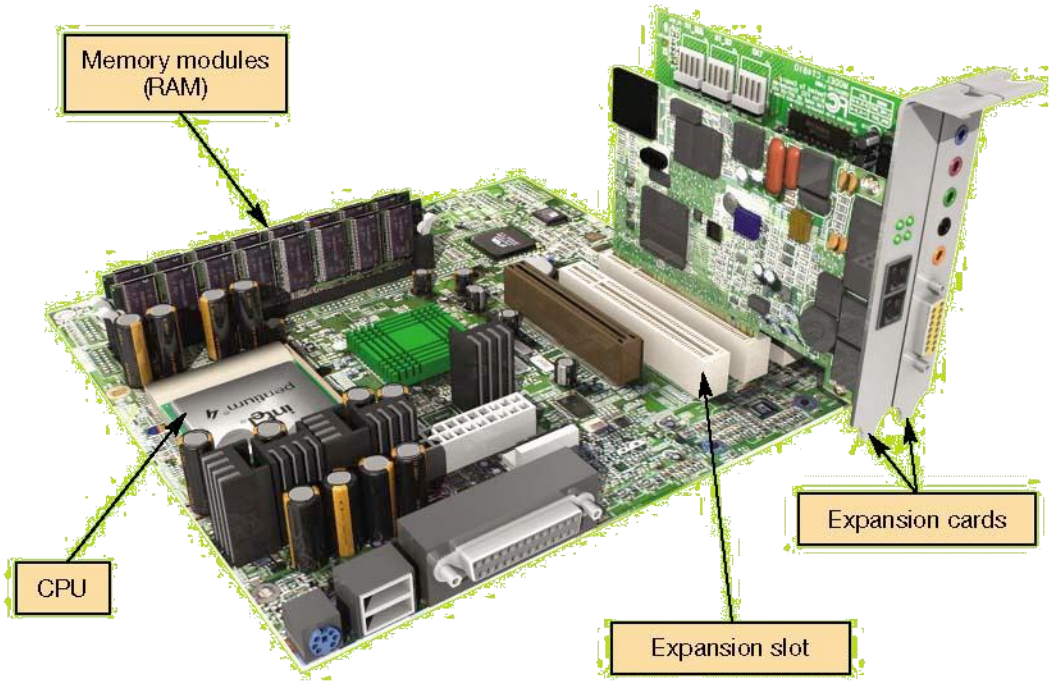
- Prej çka përbëhet një kompjuter?
  - Procesori (CPU)
  - Njësitë Hyrëse/Dalëse (I/O Units)
  - Memorja (Memory)
- Njësitë Hyrëse. (Për çka nevojiten)
  - (futjen e të dhënave)
  - Cilat janë?
    - Keyboard
    - Scanner
    - Mouse

SD/Programming

- njësitë dalëse
  - Cilat janë?
    - Softcopy
      - Video
      - Sounds
      - Sinjale controlluese
    - Hardcopy
      - Print

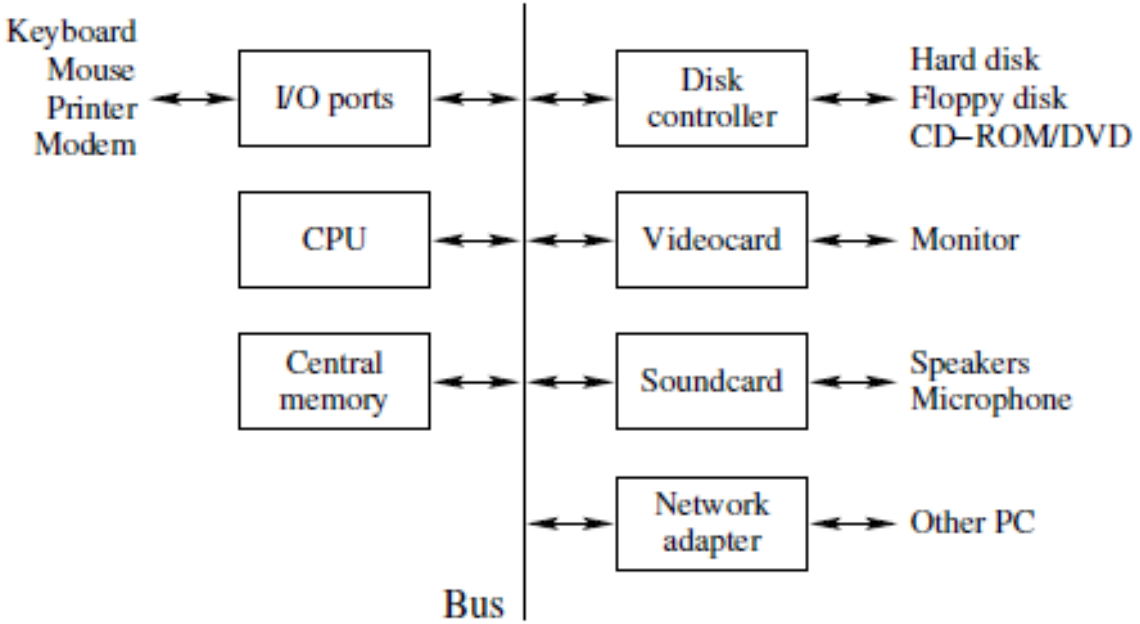
programming

- Motherboard. Çka është?
- CPU
- RAM
- Expansion Cards



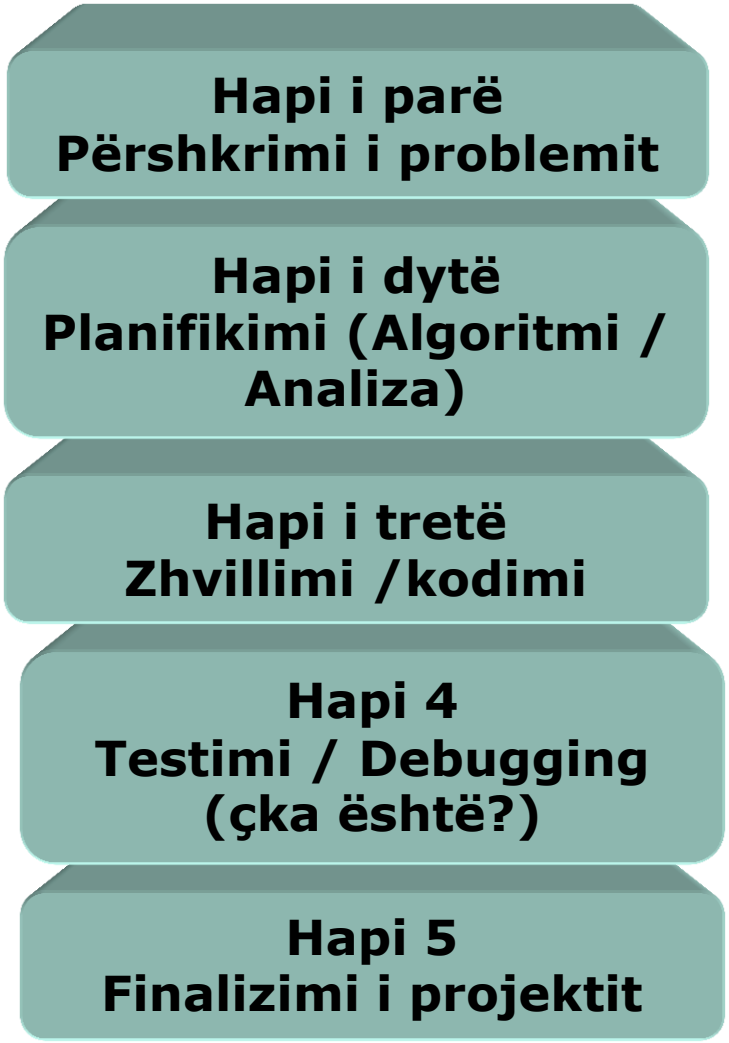
programming

- Arkitekturë e thjeshtëzume e kompjuterit



programming

- Cikli i zhvillimit të një programi.
- Sipas juve cilat hapa janë?

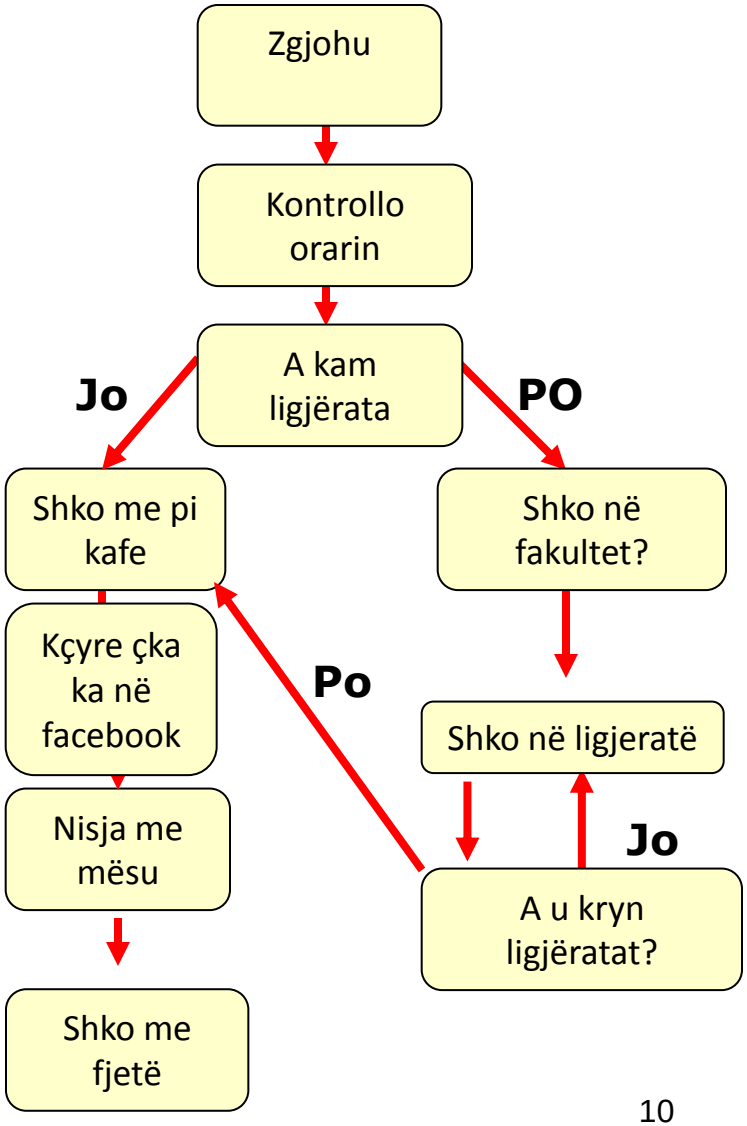


Hapi i parë – përshkrimi i problemit

Caku i programit:h	Kalkulimi i rrogës për një punëtorë të kompanisë «Happy programming».		
Të hyrat (Inputs):	Numri i orëve të punuara.....ë numër pozitiv		
Të dalat (OUTPUTS):	Paga e kalkuluar ..... Një numër pozitiv		
PROCESI:	Rroga totale kalkulohet me 8.5 EUR për orët e rregullta të punës. Secila orë shtesë mbi orarë të rregullt kalkulohet me 12.5 EUR.		
Në rast gabimi (ERROR HANDLING):	Të dhënat hyrëse duhen të jenë numra pozitiv dhe real. Nësi ipet ndonjë numër negativ apo ndonjë karakter tjetër i paprocesueshëm programi do të kërkoj dhënjen e sërishtme të të dhënave.		
Plani i testimit:	**INPUT**	**OUTPUT**	**komente**
	8	8*8.5	Testing positive input
	3	3*8.5	Testing positive input
	12	Sa? 8*8.5+4*12/5	Testing overtime input
	−6	Mesazh për dhënje të gabueshme. Kërko dhënje tëre	

Hapi i dytë – zhvillimi i një algoritmi

- Zhvillimi i algoritmit:
  - Një bashkësi e hapave të veçantë sekuencial (çka është?) që përshkruajnë se çka duhet të bëj programi
  - Algoritmet komplekse përmbajnë degëzime në vendim:
    - Binare (po/jo)
    - Loop (repeating actions)



Hapi i dytë – zhvillimi i një algoritmi

- Flow Charts:

Simbolet per operacione / procese

	Simbolizon nje proces. Simboli më i përdorueshem në flow chart

Simbolet per rrjedhe te procesit

	Tregon dretjimin e procesit
	Term inator. Simbolizon fillimin dhe perfundimin e flow chartit
	Vendim. (po/jo) degezon rrjedhen e procesit ne po/jo vija (if konstrukt)

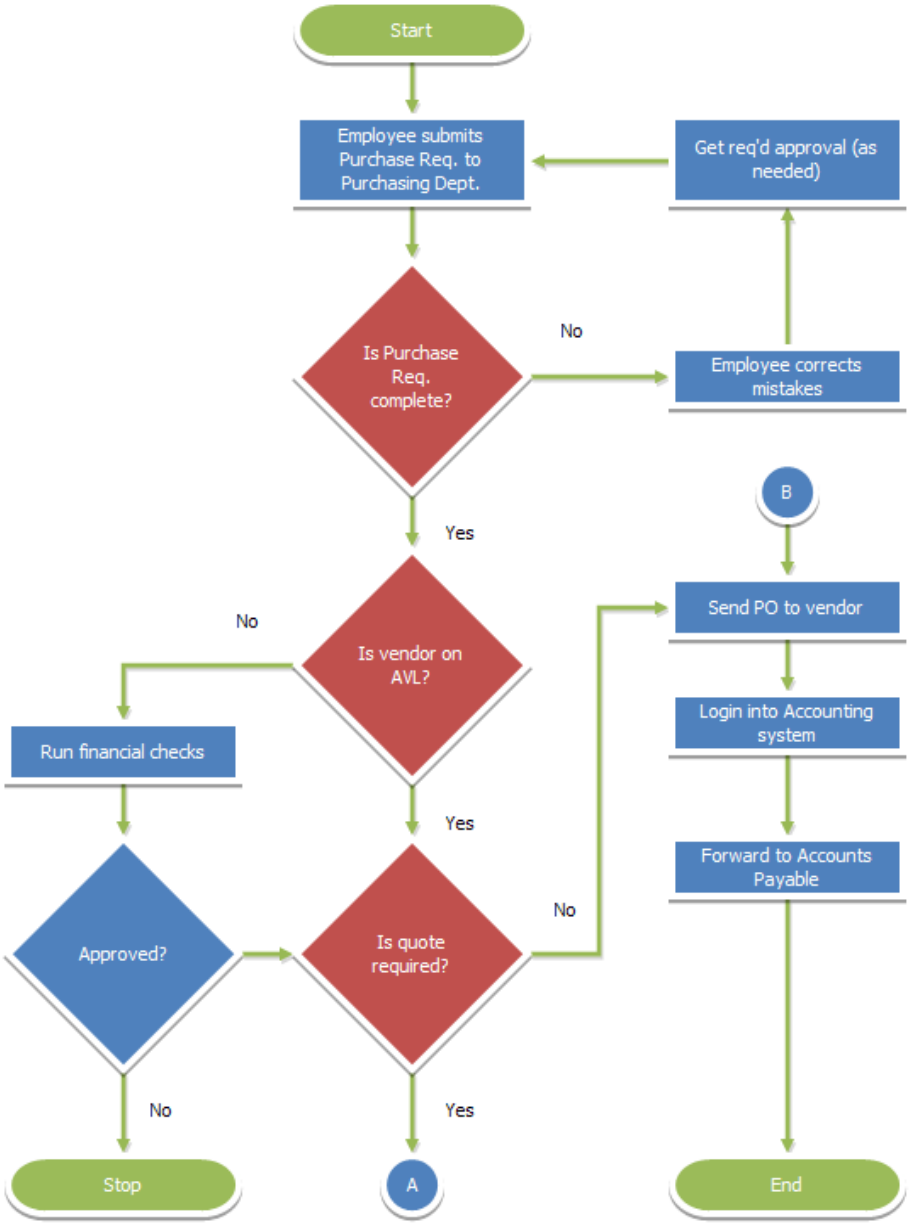
Hapi i dytë – zhvillimi i një algoritmi

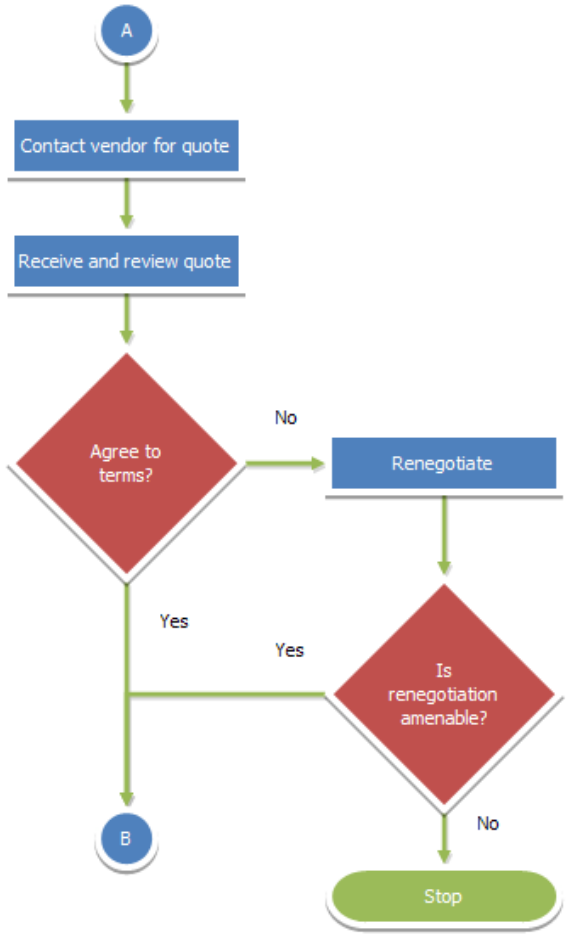
Simbolet per rrjedhe te procesit

	Konektor – lidhës. Lidhë dy pika në rrjedhën e procesit. Mund të përdorët për me thjeshtësu kompleksitetin e flow chartit. Zakonisht përdorën shkronja të mëdha të shtypit

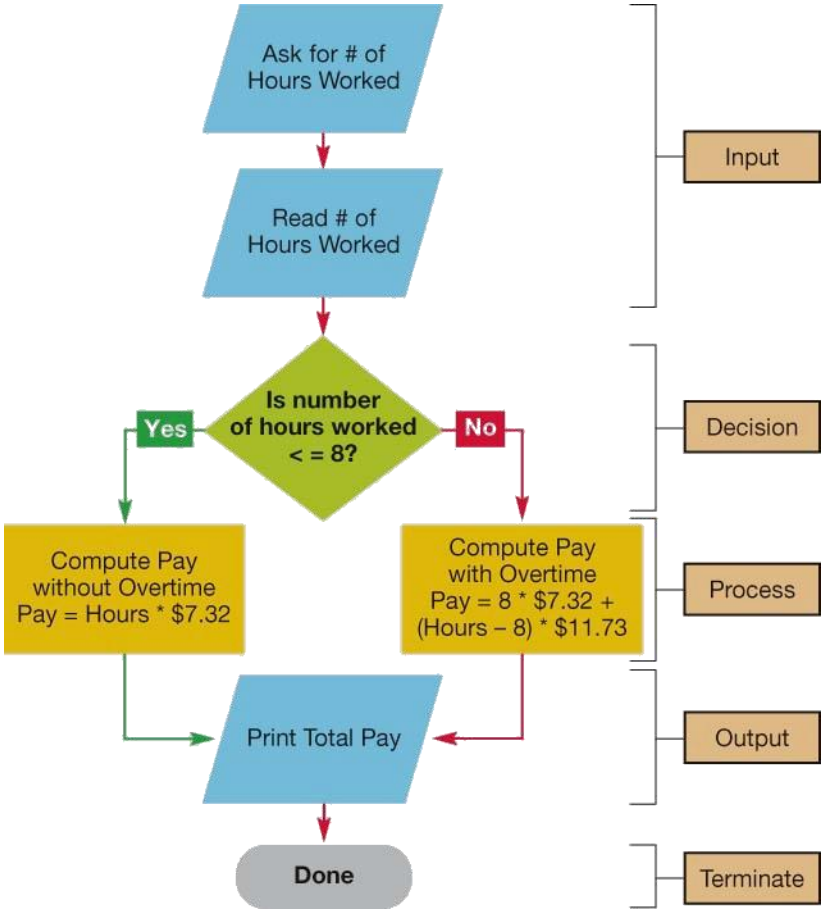
Simbolet per hyrje dhe dalje (input/output)

	Te hyrat dhe te dalat ne një proces.
	Dokument. Proces i cili si rezlutat e ka nje dokument.


Start
Employee submits Purchase Req. to Purchasing Dept.
Get req'd approval (as needed)
Is Purchase Req. complete?
No
Employee corrects mistakes
Yes
Is vendor on AVL?
No
Run financial checks
Yes
Approved?
Is quote required?
No
Yes
No
Stop
A
B
Send PO to vendor
Login into Accounting system
Forward to Accounts Payable
End

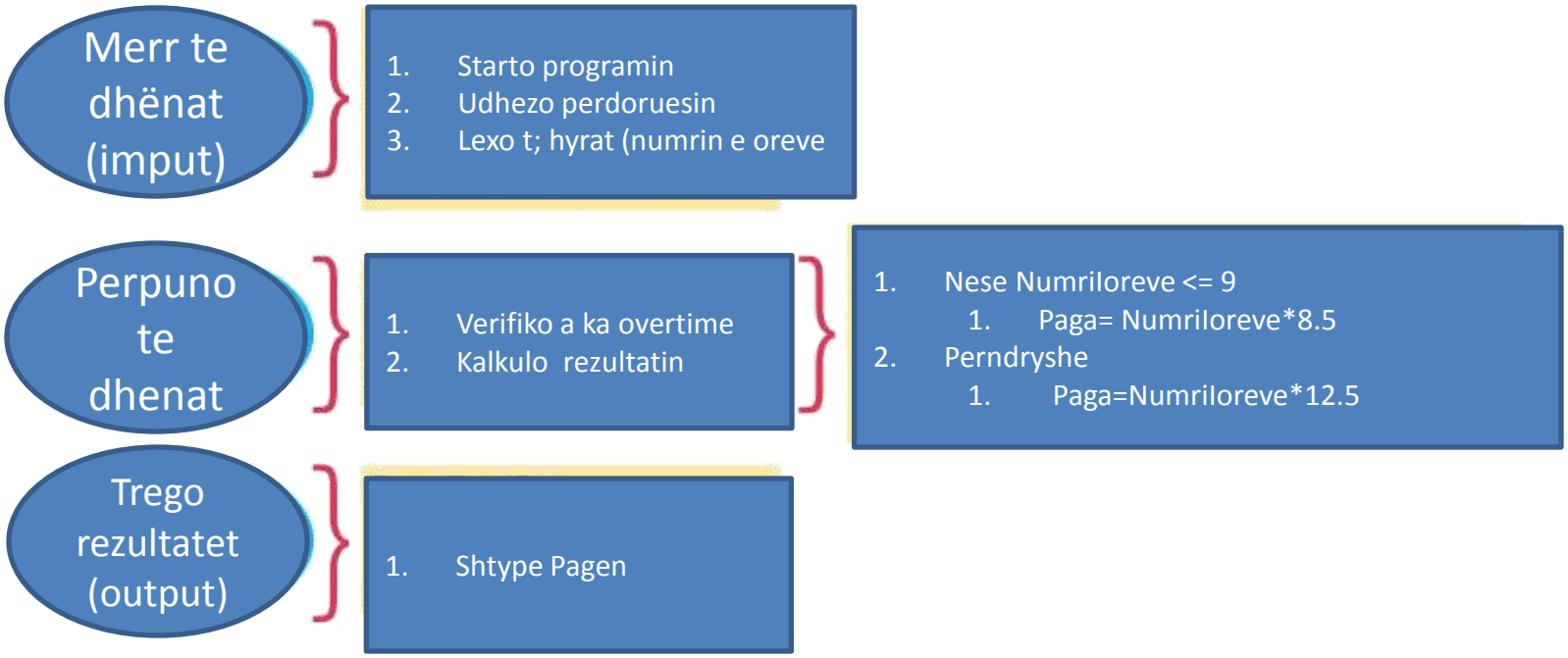

A
Contact vendor for quote
Receive and review quote
Agree to terms?
No
Renegotiate
Yes
Yes
Is renegotiation amenable?
No
B
Stop

programming



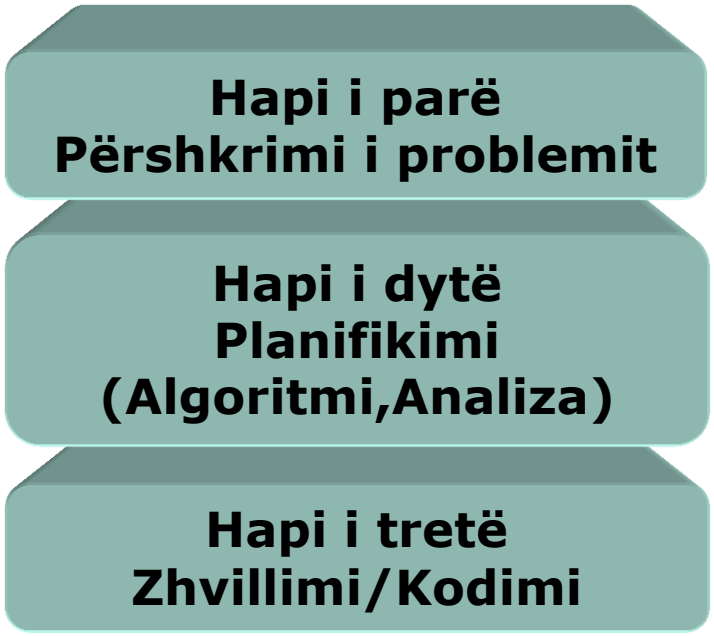
programming

- Top Down Design
- Problemi ndahet në një varg probleme gjenerale (të nivelit më të lartë)
- Detyrat e veçanta krijohen nga këto probleme më gjenerale



Programimi

- Kodimi
  - Propozone një definicion?
    - Paqyrim i një algoritmi në një gjuhë programuese.



Kodimi

- Gjenerata e parë e gjuhëve programuese
  - Gjuha e makinës (1 dhe 0)
    - 0001 1010 1100 1111
- Gjenerata e dyte e gjuhëve programuese
  - Asembleri
  - Komandat përshkruhen me fjalë
  - ADD X, Y ose MOV Y, 13
  - Duhet një përkthyes (asembleri=>ne gjuhe te makines)
- Gjenerata e tretë
  - Basic, Fortran, Java, C#
  - Më afër mënyrës se si mendojnë njerëzit
  - RrogaTotale = Orët * PagesaPërOrë
  - Përdoren simbolet (shprehje matematikore) për përshkrim të komandave.
  - Duhet një përkthyes (compiler)

Kodimi

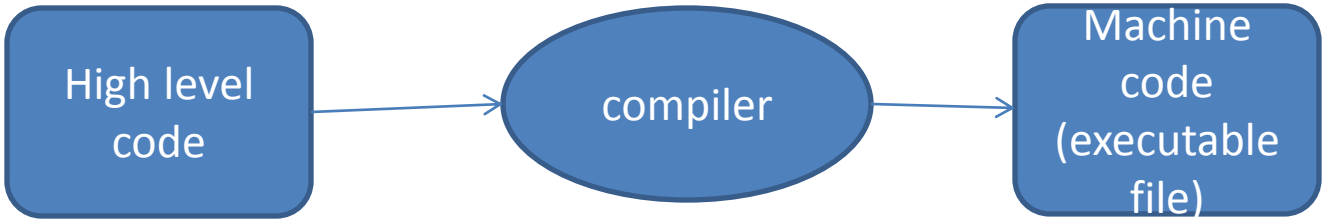
- Gjenerata e Katërt
  - SQL
  - Edhe më afër komunikimit njerëzor. Përdoren fjalë për me përshkru komanda
  - Select all where Statusi = KY and PagesaPerOre > 5

Kodimi

- Me secilën gjeneratë të gjuhëve programuese
  - Kërkesat në kompjuterë janë RRITE
  - Kërkesat në njerëz janë zvoglu

Kompajlimi

- Compiler / Kompajleri
- Çka është?
- Proces përkthimi, prej një gjuhe të lartë në gjuhë më të ulët. (zakonisht nga gjuha e lartë në gjuhë të makinës)



Kodimi

- Sintaksa dhe Semantika
- Çka janë?
- Sintaksë:
  - Mirret me rregullat e një gjuhe, si krijohen/shkruhen fjalët, pikat, presjet etj...
  - «Syntax Error»
- Semantika
  - Mirret me kuptimin e të shprehurit në një gjuhë.
  - Çka bën kompjuteri kur një urdhër ekzekutohet.

Kodimi

- Sintaksa mundet me qenë mirë, po semantika keq.
- ```
  if NumriIoreve<=8 then
     Rroga=0
  Else
     Rroga=NumriIoreve*8.5
  End if
  ```
  ```

programming

- pseudocode
- Variablat, Fjalet e rezervuara

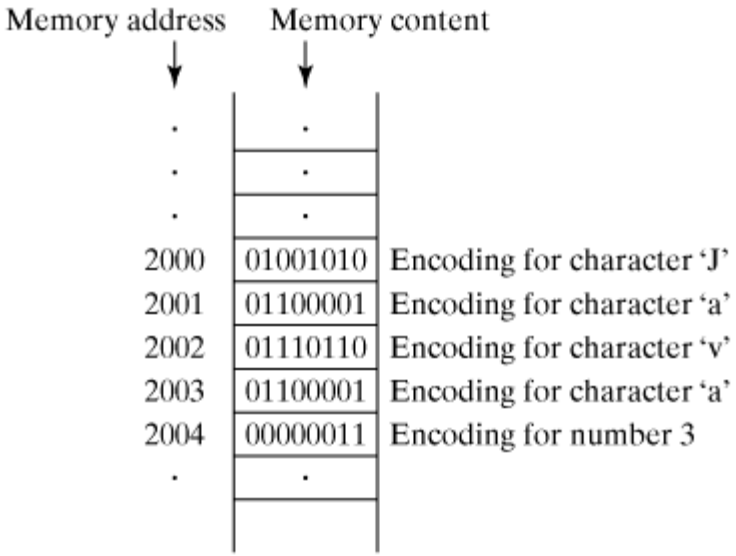
Variablat/ndryshoret

- Te dhenat, ➔ ruhen ne variabla
- Variablat ➔ copa te tedhenave në memorje
- Variablat mundësojnë ruajtjen e **përkohëshme** të të dhënave në një program.

Shënim: variablat nuk janë persistente. Kur të mbaron ekzekutimi i programit të dhënat humben. Të dhënat duhet të ruhen në një medium

variablat

- Te dhenat, ➔ ruhen ne variabla



variablat

- Secila variabël e ka
  - Emrin
  - Tipin

Tipi i variablës

Tipi i variablës e përcakton natyrën e të dhënave që përmban ajo variabël (numër, Tekst...)

Java Tipet (primitive)

- integers: tipi më i thjeshtë në Java. Numrat e plotë pozitiv dhe negativ 5, 25, -777, 1. (Java ka disat integer tipe)
- Floating point: Përdoret për ruajtjen e numrave real (me presje decimale), 3.14, 10.25. (Java ka dy floating point tipe)
- chars: përdoret për ruajtjen e karaktereve të veçanta individuale, e.g. 'c', 'e', '1', '\n'
- boolean: të dhëna logjike.

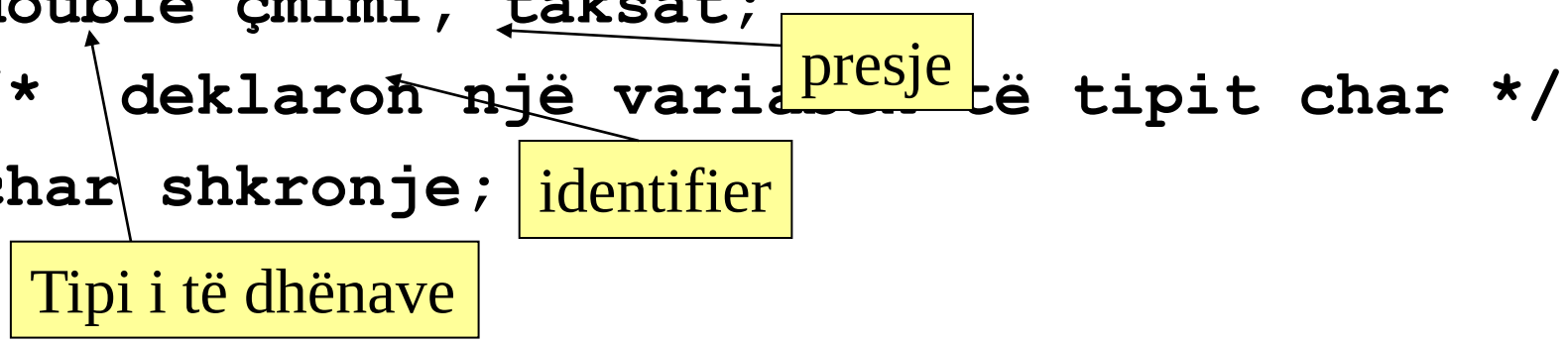
Variablat

- Emri i variablës është ekuivalent me adresën e variablës në memorje
- Secila variable e ka një **emër**, **tip**, dhe **vlerë**
- Kur një vlerë vendoset në një variabël (adresë në memorje) ajo vlerë e zëvendëson vlerën e kaluar (*destructive read-in*)
- Vlera e një variable mund të lexohet pa u ndryshuar / shkatërruar (*non-destructive read-out)*

Deklarimi i variablave

- Para se me përdorë një variabël ajo duhet të deklarohet. (Jo të gjitha gjuhët programuese e kërkojnë këtë send, mirëpo Java, C, C++, C# po)
- Shembuj (në java):

```
/*  Deklaron një variabël integer */
int numer;
/*  Deklaron dy variabla reale */
double çmimi, taksat;
/*  deklaron një variabël të tipit char */
char shkronje;
```



Tipet numerike ne java

Integer

byt e	8 bits	-128	127
shor t	16 bits	-32768	32767
i nt	32 bits	-2,147,483,648	2,147,483,647
l ong	64 bits	-9,223,372,036,854,770,000	9,223,372,036,854,770,000

Floating point

float	32	-3.40282347E+38	3.40282347E+38
double	64	-1.79769313486231570E+308	1.79769313486231570E+308

Operacionet me variabla

- Deklarimi i operacioneve përcaktuese

```
int x;
int y;
int z;
x = 5;
y = 10;
z = x + y;
```

- Inicializojmë x, z dhe ruajmë vlerën e shumës së tyre në variablën z

Operacionet bazike matematikore

Java Operation	Arithmetic Operator	Algebraic Expression	Java Expression
Addition	+	a + b	a + b
Subtraction	-	a – b	a – b
Multiplication	*	ab	a * b
Division	/	a / b	a / b
Modulus	%	a mod b	a % b

Operacionet me variabla

- Ta zëmë që kemi kodin në vijim

```
int x;
x = 7 / 4;
```

- A ka ndonjë koment në këtë kod?
- Me kalkulator rezultati është 1.75
- Mirëpo x është nuk është integer.

Prioriteti i operacioneve

- Sa eshte rezultati?

```
x = 7 + 3 * 6;
```

- Dy mundesi:

  → 7 + 3 = 10 → 10 * 6 = 60

  → 3*6 =18 → 7+18 = 25

- Cili verzion asht korrekt?

Prioriteti i operacioneve

- Rregullat e evaluimit te operacioneve matematikore
- Rregullat jane te ngjashme ne shumicen e gjuheve programuese.

Operator(s)	Operation(s)	Order of evaluation (precedence)
`()`	Parentheses	Evaluated first. If the parentheses are nested, the expression in the innermost pair is evaluated first. If there are several pairs of parentheses "on the same level" (i.e., not nested), they are evaluated left to right.
`*`, `/`, or `%`	Multiplication Division Modulus	Evaluated second. If there are several, they are evaluated left to right.
`+` or `-`	Addition Subtraction	Evaluated last. If there are several, they are evaluated left to right.

Vlerat Boolean

- Sipas George Boole, nje matematicienti Anglez i cili ne vitin 1854 ka botu librin “An investigation into the Laws of Thought”. Liber qe njifet si filli i logjikes se boolit.
- Secila shprehje e boolit rezulton ne `true` ose `false`. (e vertete/e pa vertete)
- Java e ka tipin boolean

Operatoret boolean

Operatori	Kuptimi
>	Me i madh se
<	Me i voges le
>=	Me i madh ose i barabarte se
<=	Me i vogel ose i barabarte se
==	i barabarte me
!=	i ndryshem me

Shembull

```
public void showBool()
  {

     boolean boolVar;

            boolVar = false;
            System.out.println ("boolVar: " + boolVar);

            int a = 10;
            int b = 10;
            boolVar = ( a == b);
            System.out.println ("boolVar: " + boolVar);

            System.out.println (a == b);
            System.out.println (a != b);
            System.out.println (a < b);
            System.out.println (a <= b);
            System.out.println (a > b);
            System.out.println (a >= b);

  }
```

```
boolVar: false
boolVar: true
true
false
false
true
false
true
```

- Kujdes

```
( temperatura = 100 )
( temperatura == 100 )
```

nuk eshte njesoj.

Degezimet - `if`

- Pseudocode:

```
if some Boolean expression is true
   do this
```

- Shembull:

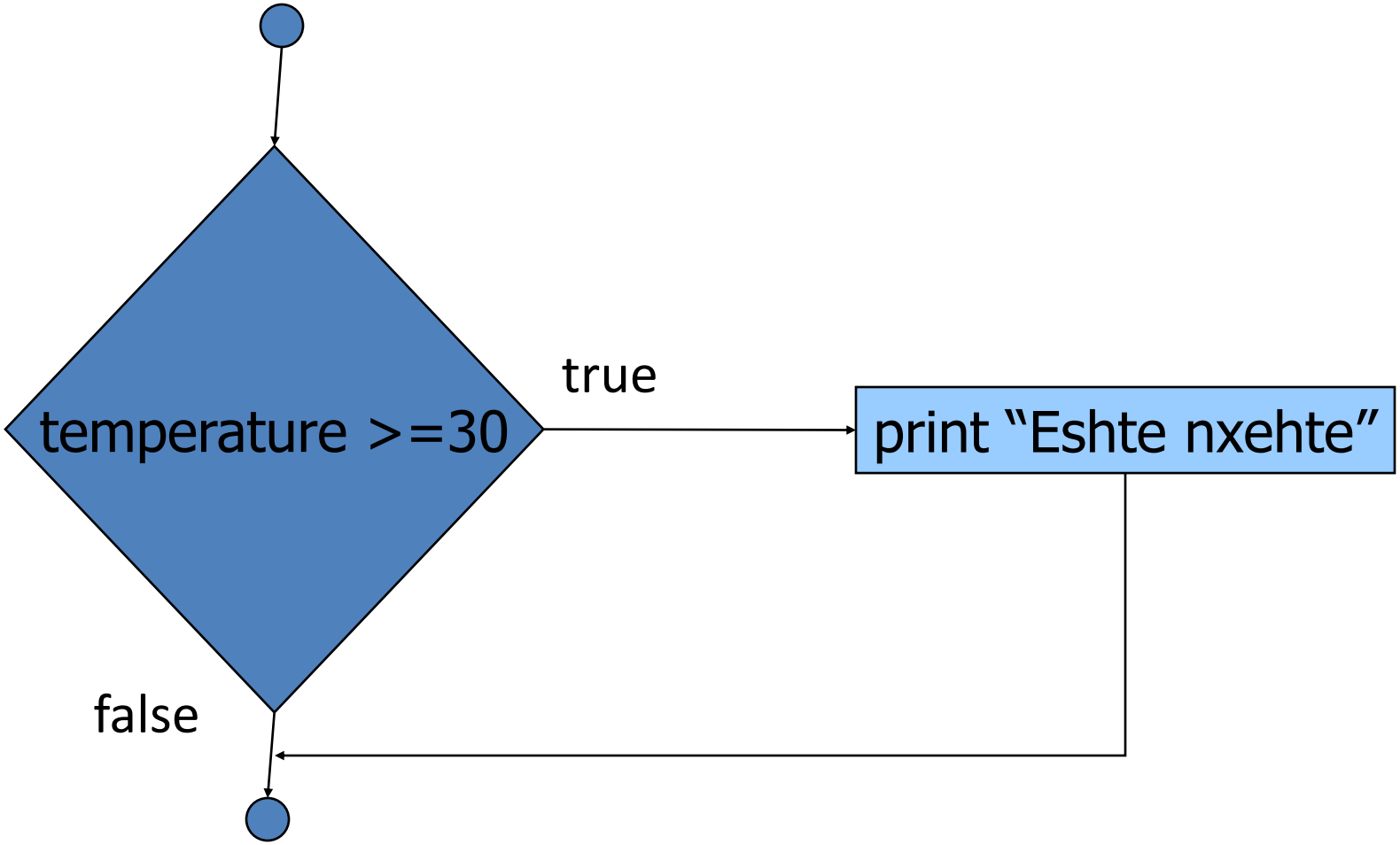
```
if ( x == y )
{
   System.out.println(" x is equal to y!" ) ;
}
```

- Secila gjuhe procedurale/OO e ka kete strukture.

Nje shembull tjeter `if`

```
if ( temperature >= 30 )
{
 System.out.println( “Eshte nxehte" );
}
```

`if` Flow Chart



`if` me alternative: `if else`

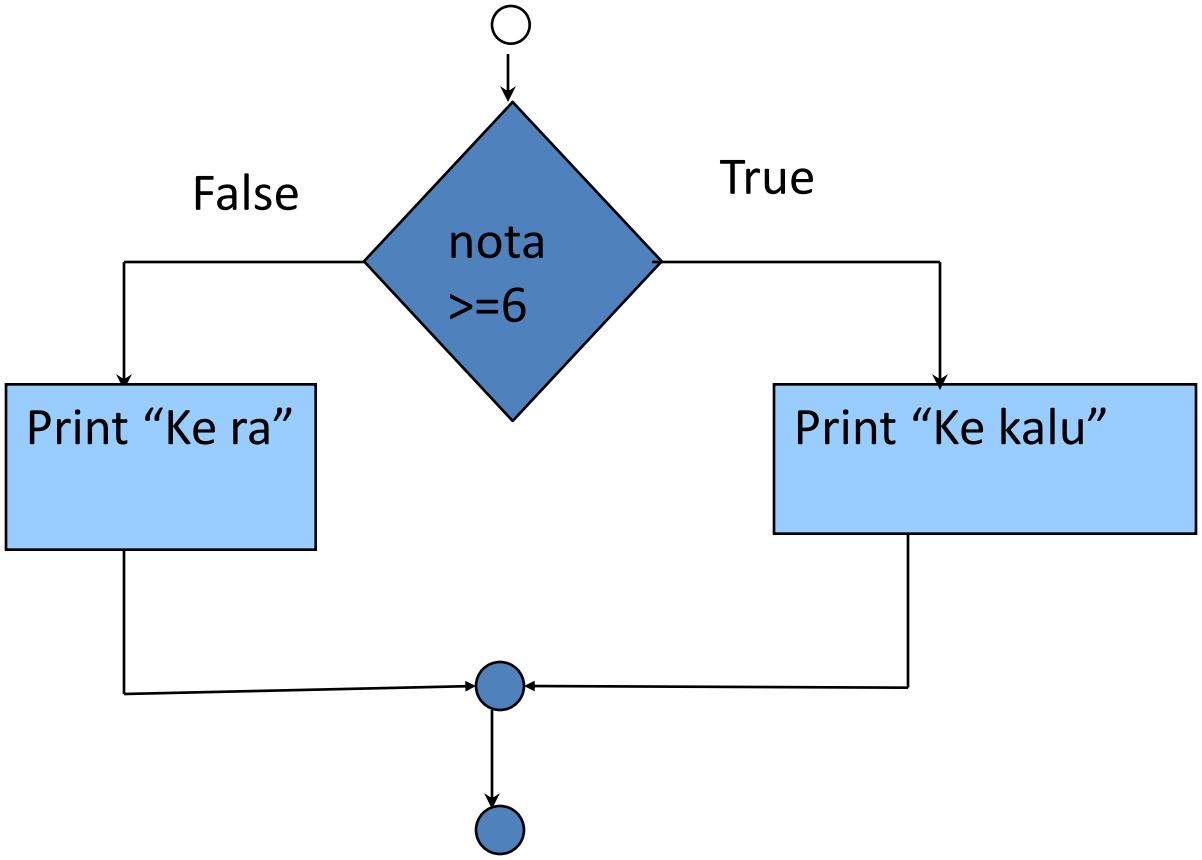
- Pseudocode:

```
if some Boolean expression is true
   do this
otherwise
   do something else
```

- Example:

```
if ( nota >= 6 )
     System.out.println( “Ke kalue!" );
else
     System.out.println( “Ke ra!" );
```

`if/else` Flow Chart



Bloqet

- Per me ekezekutu ma shume urdhera duhet me i paketu me nje bllok
- P.sh.

```
if ( nota >= 60 )
{
     System.out.println ( “Urime" );
     System.out.println ( “Ke Kalu!" );
}
```

Formatimi

- Mos programo keshtu:

```
if (grade >= 65)  {
System.out.println("You passed!!!\n");
System.out.println ("Congratulations!\n");
}
```

- Eshte kod valid mirepo nuk eshte i lexueshem dhe veshtireson mirembajten - *bad style*

Gabim i zakonshem: pike-presja

- Ne java secila komande duhet te permbyllet me pikepresje.
- Sintaksa e IF:

```
if (boolean expression)
   Statement or block of code;
```

- Gabim

```
if (boolean expression);
  Statement or block of code;
```

Shembull

```
public class Nota
{

    public void Noto(int nota)
    {

          /* find out if grade is passing */
          if ( nota >= 6 )
          {
                    System.out.println ( "Urime!!!" );
                    System.out.println ( "Ke Kalu!" );
          }
          else
          {
                    System.out.println ("Ke ra!");
          }

    }
}
```

Operatori “dhe - && - AND”

```
((boolean exp a) && (boolean exp b))
```

- Kur perdoret operatori dhe (&&), te dy gjykimet a dhe b duhet me qene te verteta qe shprehja me qene e vertete.

	FALSE	TRUE
FALSE	FALSE	FALSE
TRUE	FALSE	TRUE

- Shembull:

```
((total > 50) && (status == 0))
```

Operatori “ ose -||-or “

```
((boolean exp a) || (boolean exp b))
```

- Kur perdoret operatori dhe (||), njera prej gjykimeve a ose b duhet me qene i vertete qe shprehja me qene e vertete.

truth table

	FALSE	TRUE
FALSE	FALSE	TRUE
TRUE	TRUE	TRUE

- Shembull:

```
((total > 50) || (status == 0))
```

^ - Exclusive OR

```
((boolean exp a) ^ (boolean exp b))
```

- Operatoret duhet ti kene vlerat e kunderta ne menyre qe shprehja te jete e vertete

	FALSE	TRUE
FALSE	FALSE	TRUE
TRUE	TRUE	FALSE

- Shembull

```
((person1 == 1) ^ (person2 == 1))
```

perseritje

- Si eshte dalja (output) e kodit ne vijim?

```
int a = 100;
if (a != 100);
   System.out.println (" a eshte" + a);
```

- Si eshte dalja (output) e kodit ne vijim?

```
int a = 100, b = 1;
if (a < b)
   System.out.println (“a me e vogel se
    b");
   System.out.println ("Thank you");
```

perseritje

Prioriteti i operatoreve

- a + b * (c + 10 * d) / e
- 3 + 4 * 4 > 5 * (4 + 3) – 1

Nga Libri (liang, kap. 3.12)

- Is 10 divisible by 5 and 6? false
- Is 10 divisible by 5 or 6? true
- Is 10 divisible by 5 or 6, but not both? true

Komanda - Switch

- Perdoret vetem ne rastin kur nje ndryshore ose nje shprehje testohet per barazi (d.m.th. Jo per >, <, >=, <=) me secilen vlere te plote qe mund te marre ajo .
- Me e pershtatshme se if/else ne raste kur testohet nje shprehje per barazi me vlera te ndryshme

switch

```
switch (expression) {
   case value1:
      action(s);
      break;
   case value2:
      action(s);
      break;
   …
   default:
      actions(s);
      break;
}
```

Fjala kyce - switch

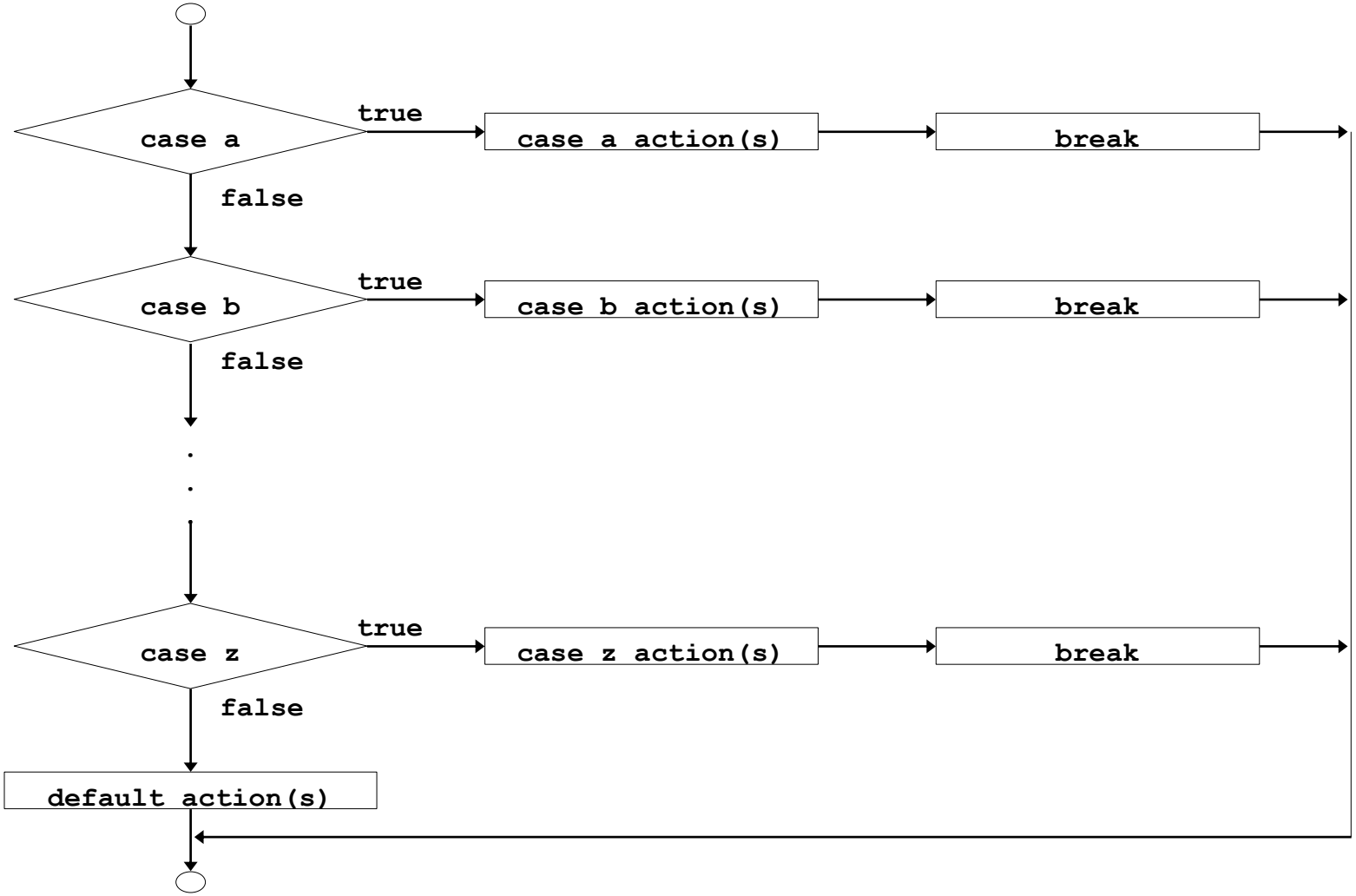
expression mund te jete ndryshore
Apo nje shprehje me e nderlikuar

Per secilen vlere qe testohet nje case

Urdherat brenda nje blloku nuk
Kane nevoje per kllapa

default case –ekzekutohet ne rastin
Kur nuk plotesohet asnje case

switch


case a
true
case a action(s)
break
false
case b
true
case b action(s)
break
false
case z
true
case z action(s)
break
false
default action(s)

Kujdes

- Nese harrohet “break” te gjithe urdherat do te ekzekutohen deri te break i ardheshem.

shembull

- Sa eshte dalja (output) I shembullit ne vijim?

```
int a = 90;
switch (a)
{
 case 80:
   System.out.println (“Test1”);
   break;
 case 90:
   System.out.println (“Test2”);
   break;
 case 100:
   System.out.println (“Test3”);
   break;
 }
```

ushtrim

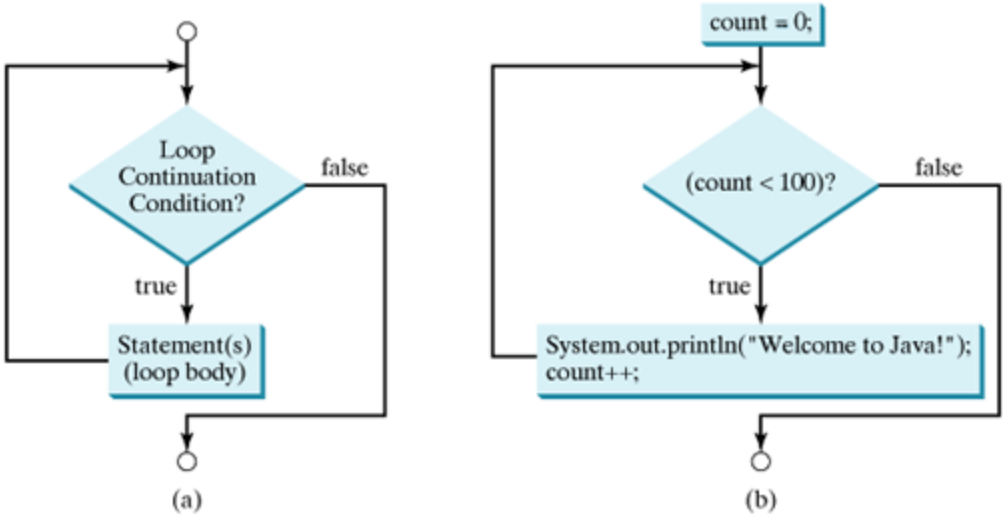
- Te shkruhet nje program i cili e shqipton noten (per 1 – dobet, 2 – Mjaftueshem, 3- Mire, 4- Shume Mire, 5 – Shkelqyeshem)

Loops – Unazat, Laqet, Ciklet

- While Loop (deri sa, gjersa)
- Perserite nje process deri sa plotesohet nje kusht.

(pseudocode)

```
while (some Boolean expression is true)
{
        do this (again and again...)
}
```



shembull

- Shtype 15 here “Hello Prizren”

```
int numri = 0;
while (numri < 15) {
        System.out.println(“Hello prizren");
        numri=numri+1;
}
```