

Pachete

→ organizarea unui program în subsisteme

package no. opt. curs --- idk

unități de compilare
(fișiere)

pachet

import nume Pachet . nume Clasă ;

import nume Pachet . * ; → toate clasele
din pachet

conflict de nume → clasa importată
specific (nu cu *)
are prioritate

import clocks . * ;
import aux . * ;

clocks . Clock / aux . Clock

include ≠ import
↑ (C) (java)

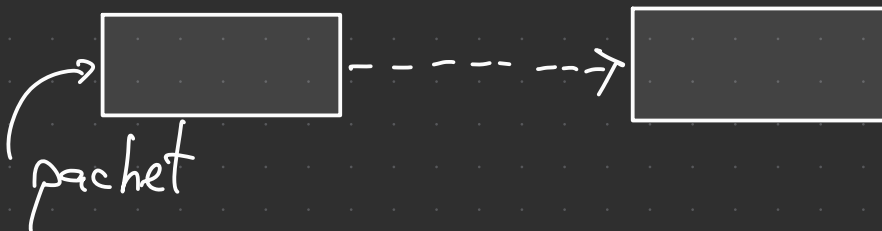
se înlocuiește
cod

import static ... *
→ membrii statici !!!

import java.lang.*;

-> adăugat automat în fiecare unitate de compilare

Dependențe



vizibilitate în pachete (a claselor)

public -> accesabil și din ext. pachetului

(clasa trebuie să aibă același nume cu fișierul)

interfața pachetului

default (fără niciun cuvânt cheie)

-> doar în pachet (detaliu de implementare a pachetului)

(modificatori acces membrii)

vizibilitate

private
public
protected
default

-> package access (doar în pachet)

~ UML

Compiler si Rulare

*.java $\xrightarrow{\text{javac}}$ { *₁.class
 *₂.class

bin

└ folder nume 1

└ folder nume 2

└ folder nume 3

└ x.class

└ y.class

package nume1.nume2.nume3;

class x

class y

javac -d calc -cote_bin *.java