

Informatii si termeni de baza in JAVA

Informatie/termen	Explicatie
Denumirea fisierului <i>.java</i>	Fisierul <i>.java</i> trebuie sa aiba numele identic cu numele clasei
javac <i>numeFisier.java</i>	Comanda de compilare a unui fisier <i>java</i> in linie de comanda
java <i>numeFisier</i>	Comanda de lansare in executie a unui program in linie de comanda (se va executa <i>numeFisier.class</i> creat in urma compilarii corecte)
Tipuri de date primitive	<ul style="list-style-type: none"> • intregi: <i>byte, short, int, long</i> • reale: <i>float, double</i> • caracter: <i>char</i> • logic: <i>boolean</i>
Declararea unei variabile	<code>int a;</code>
Initializarea unei variabile	<code>a = 2;</code>
Definire	Declarare + Initializare
Citirea de la tastatura	<p>Pasul 1: crearea unui obiect de tip <i>Scanner</i></p> <p><i>Scanner keyboard = new Scanner(System.in);</i></p> <p>Pasul 2: folosirea obiectul pentru citirea datelor necesare</p> <p>sir de caractere -> <i>keyboard.nextLine();</i></p> <p>int -> <i>keyboard.nextInt();</i></p> <p>float -> <i>keyboard.nextFloat();</i></p> <p>double -> <i>keyboard.nextDouble();</i> etc.</p>
Afisarea unui mesaj in consola	<p><i>System.out.println("mesaj");</i> // afisare cu trecere la linie noua</p> <p><i>System.out.print("mesaj");</i> // afisare fara trecere la linie noua</p>
Instructiuni de decizie/selectie	<ul style="list-style-type: none"> • <i>if</i> – evaluarea unei conditii (poate fi folosit in functie de necesitate impreuna cu clauzele <i>else</i> si <i>else if</i>) • <i>switch</i> – solutie alternativa pentru <i>if-else</i>, compara o expresie cu mai multe valori date <ul style="list-style-type: none"> • in cazul in care dupa o clauza <i>case</i> nu se regaseste comanda <i>break</i>, atunci se vor executa si cauzele <i>case</i> urmatoare pana la intalnirea unei instructiuni <i>break</i> / pana la terminarea <i>switch</i>-ului

	<ul style="list-style-type: none"> • cauza implicita <i>default</i> se poate regasi dupa ultima clauza <i>case</i>
Instructiuni iterative	<ul style="list-style-type: none"> • <i>for</i> – instructiune iterativa cu test initial si numar cunoscut de pasi !! Dupa parantezele instructiunii <i>for</i> NU se pune ; • <i>while</i> - instructiune iterativa cu test initial si numar necunoscut de pasi !! Dupa parantezele instructiunii <i>while</i> NU se pune ; • <i>do-while</i> - instructiune iterativa cu test final si numar necunoscut de pasi !! Instructiunile din blocul de instructiuni se vor executa cel putin o data, chiar daca expresia este falsa
Instructiuni de salt	<ul style="list-style-type: none"> • <i>break</i> – iese din blocul de instructiuni • <i>return</i> – oprirea executiei programului • <i>continue</i> – sare la pasul urmator
Declararea unui tablou unidimensional	<code><tip_elemente> <identificator_tablou>[];</code> sau <code><tip_elemente>[] <identificator_tablou>;</code>
Declararea si instantierea unui tablou unidimensional	<code><tip_elemente> <identificator_tablou>[] =</code> <code>new <tip_elemente>[<nr_elemente>;</code> !! La declararea unui tablou se specifica intotdeauna numarul de elemente !! Primul index intr-un tablou este 0 , iar ultimul este <i>nr_elemente-1</i>
Declararea si instantierea unei matrici	<code><tip_elemente> <identificator_tablou>[][] =</code> <code>new <tip_elemente>[<nr_linii>][<nr_coloane>;</code>
Numarul de elemente dintr-un tablou	proprietatea <i>length</i>
Clasa <i>Arrays</i>	<ul style="list-style-type: none"> • se foloseste pentru solutii rapide in lucru cu tablouri • se regaseste in pachetul <i>java.util</i> • metode des utilizate: <i>equals()</i>, <i>fill()</i>, <i>sort()</i>, <i>binarySearch()</i>
Siruri de caractere	<ul style="list-style-type: none"> • <i>String</i>, <i>StringBuffer</i>, <i>StringBuilder</i> • Metode des utilizate in lucru cu <i>String</i>-uri: <i>compareTo()</i>, <i>equals()</i>, concatenare(+), <i>charAt()</i>, <i>replace()</i>, <i>substring()</i>, <i>indexOf()</i>, <i>lastIndexOf()</i>, <i>toUpperCase()</i>, <i>toLowerCase()</i>, <i>split()</i>, <i>trim()</i> • Metode specifice <i>StringBuffer/StringBuilder</i>: <i>append()</i>, <i>insert()</i>, <i>delete()</i>, <i>reverse()</i>, <i>toString()</i>

POO / OOP	Programare orientate pe obiecte / Object Oriented Programming <ul style="list-style-type: none"> • Presupune crearea unor tipuri de date complexe asemanatoare obiectelor din viata reala
Denumire clase	Prin conventie numele claselor incepe cu majuscule
Instantiere	Obtinerea unui obiect care are propria locatie in memorie si propriul cod
Mostenire	Mecanism de obtinere a unor clase din clase deja existente <ul style="list-style-type: none"> - subclasa (clasa copil) mosteneste proprietatile si metodele (public si protected) ale superclasei (clasei parinte) - Regula: “este-un” / “este-o” - Este unidirectionala - Cuvant cheie: <i>extends</i> <pre>public class Subclasa extends Superclasa { ... }</pre>
Incapsulare	Mentinerea datelor si metodelor separate de exterior
Abstractizare	Ascunderea anumitor functionalitati
Polimorfism	Mecanism care permite accesare proprietatilor / metodelor unui obiect prin intermediul unor tipuri de date , obtinut prin derivare (mostenire)
Semnatura unei metode	Nume + Tipul parametrilor
Modificatori de acces ai claselor	<ul style="list-style-type: none"> • <i>public</i> • <i>implicit</i>
Modificatori de acces ai proprietatilor/metodelor	<ul style="list-style-type: none"> • <i>public</i> • <i>protected</i> • <i>implicit</i> • <i>private</i>
Modificatorul <i>final</i>	<ul style="list-style-type: none"> • aplicat unei proprietati => constanta • aplicat unei metode => metoda NU poate fi supradefinita in subclasa • aplicat unei clase => clasa NU poate fi mostenita
Modificatorul <i>static</i>	<ul style="list-style-type: none"> • daca o proprietate/metoda are modificador <i>static</i> => proprietate/metoda de clasa • daca o proprietate/metoda NU are modificador <i>static</i> => proprietate/metoda de instanta

Supraincarcare	Metode cu acelasi nume, dar semnături diferite
Supradefinire	O metoda din superclasa este definita ca metoda si in subclasa (cu aceeasi semnatura)
Constructori	<p>Metoda speciala care este apelata automat la crearea unei instante</p> <p>!! Constructorul nu are tip returnat si nu trebuie sa contina instructiunea return</p> <p>!! Constructorul are acelasi nume cu cel al clasei</p> <p>Daca nu este scris niciun constructor, atunci se va apela automat constructorul implicit</p> <p>Daca este definit un constructor ,indiferent de numarul de parametri, cel implicit nu va mai fi creat.</p>
Diferentierea intre variabile de instanta si variabile locale	<p>this</p> <p>Ex: <i>this.nume = nume;</i> //<i>this.nume</i> este variabila de instanta, iar <i>nume</i> este variabila locala</p>
Tipuri de erori	<ul style="list-style-type: none"> • de compilare: usor de descoperit, greseli de scriere • de executie: introducerea gresita a datelor, indisponibilitatea unei resurse
Tratarea erorilor	<pre>try{ ... } catch(<tip_exceptie>){ ... } - se poate adauga si blocul finally{ ... } care se va executa indiferent daca se produc sau nu erori</pre>
Interfete	<ul style="list-style-type: none"> - Defineste metodele, NU le implementeaza. - Implementeaza concepul de separare a modelului unui obiect de implementarea sa. <pre>public interface InterfataA{ ... }</pre> <ul style="list-style-type: none"> - Cuvant cheie: implements

	<pre>public class ClasaA implements InterfataA { ... }</pre>
Clasa <i>File</i>	<ul style="list-style-type: none"> - metode pentru lucru cu sisteme de fisier si structuri de directoare - exemple de metode: <i>isFile()</i>, <i>renameTo()</i>, <i>delete()</i>, <i>isHidden()</i>, <i>length()</i>, <i>isDirectory()</i>, <i>canRead()</i>, <i>canWrite()</i>, <i>exists()</i>, <i>getParent()</i>, <i>getPath()</i>, <i>getName()</i>, <i>list()</i>
Scrierea, citirea fisierelor	<ul style="list-style-type: none"> - exemplu fluxuri: <i>PrintWriter</i>, impreuna cu <i>FileOutputStream</i> sau <i>BufferedWriter</i>, impreuna cu <i>FileWriter</i> / <i>BufferedReader</i>, impreuna cu <i>FileReader</i> - metode: <i>print()</i>, <i>println()</i> sau <i>write()</i>, <i>newline()</i>, <i>append()</i> / <i>readLine()</i> - Atentie la inchiderea fluxurilor!
Scrierea, citirea unor obiecte in fisiere	<ul style="list-style-type: none"> - clasa trebuie sa implementeze interfata <i>Serializable</i> - exemplu fluxuri: <i>ObjectOutputStream</i>, impreuna cu <i>FileOutputStream</i> / <i>ObjectInputStream</i>, impreuna cu <i>FileInputStream</i> - metode: <i>writeObject()</i> / <i>readObject()</i> - Atentie la inchiderea fluxurilor!
Interfata grafica cu utilizatorul (GUI)	<ul style="list-style-type: none"> - Comunicare vizuala intre program si utilizator - Pachet pentru servicii grafice <i>java.awt</i> sau <i>javax.swing</i> - Casete de dialog: <i>JOptionPane</i> - Varianta utilizare: se mosteneste clasa <i>JFrame</i> - Exemple: <i>JPanel</i> (container), <i>JLabel</i> (control static de tip text), <i>JTextField</i> (control de editare), <i>JBUTTON</i>, <i>JTable</i>, <i>JScrollPane</i> etc.