

# Поделба на програмските јазици

Најчесто се користат две поделби на програмските јазици:

## 1. Поделба на програмските јазици:

- според начинот на пресметување
- според областа на примена

## 2. Поделба според начинот на пресметување:

- императивни програмски јазици
- декларативни програмски јазици

Во програмите напишани во императивни програмски јазици проблемите се решаваат т.ш. се потенцира како компјутерот ќе ги иврши пресметувањата, а кај програмите напишани на декларативен програмски јазик се потенцира што ќе пресмета.

## 3. Императивни програмски јазици

Императивен доаѓа од зборот *imperare* што значи наредува. Затоа се нарекуваат и наредбени јазици. Составени се од наредби со кои се менуваат вредностите на податоците. Во нив спаѓаат: *fortran, cobol, basic, pascal, modula, c, ada*.

Се делат на:

- процедурални
- објектно ориентирани

Со процедуралните јазици програмите се пишуваат според постапката која е опишана со избраната метода за решавање на задачата. Притоа се користат техники на т.н. **структурно програмирање** со кое задачата се дели на помали и полесни задачи. За секоја задача се користи соодветна метода за решавање со која од почетните влезни податоци преку аритметички и логички пресметувања се добиваат крајните резултати. Една метода може да се примени за различни влезни податоци. Пр. Ако радиус на кругот е 17 cm, со пресметување на плоштина на круг  $r^2\pi$  се добива неговата плоштина, но ако 17 се годините на некој ученик ( $r^2\pi$ ) со истата метода, резултатот нема никаква смисла.

Овој недостаток е решен со **објектно – ориентирани јазици**. Во нив секој објект (ученик, триаголник, автомобил, брод) има свои методи кои може да се применат на неговите податоци. Меѓу најпознатите процедурални јазици се *pascal, basic, fortran*, а објектно – ориентирани се *c++, java, Delphi, simula* и др.

## 4. Декларативни програмски јазици

Се делат на:

- функциски
- логички

Програмите во **функциските јазици** претставуваат функција (или повеќе функции) составена од попусти функции. Една функција може да повикува друга, а резултатот од функцијата може да се користи како аргумент во друга функција. Главна карактеристика е тоа што се користат изрази и функции, за разлика од императивните јазици кои користат наредби и процедури. Познати се: *lisp, ML, scheme, Miranda*

Пр. Функцијата  $F$  со аргументи  $X$  и  $Y$  во *lisp* се претставува со  $(F X Y)$ . Основните елементи кои се користат во *lisp* се наречени листи и атоми.

**Логичките јазици** се наменети за решавање на проблеми од вештачката интелигенција. Ги користат принципите на т.н. формална логика каде се задава множество на тврдења (аксиоми) и правила на заклучување. Со примена на правилата на заклучување врз тврдењата се доаѓа до одговорот на поставената задача, односно се генерираат нови тврдења и правила на заклучување. Најпознат програмски јазик е *prolog*. Во него тврдењето „Ивица е пријател на марица“ се запишува:

*Prijatel(Ivica, Marica).*

## 5. Поделба според областа на примена

Со еден програмски јазик не може да се изразат решенијата на проблеми од различни области. Пр. Во техничките проблеми има голем број на математички пресметувања, а во економските опишување и табелирање.

- деловни апликации
- научни апликации
- системски софтвер
- компјутерска комуникација
- вештачка интелигенција
- специјална намена

Вишите програмски јазици не зависат од машината на која ќе се изведуваат и затоа се наречени машински независни јазици. Затоа при пишувањето на програмата не се води сметка за типот на компјутерот на кој ќе се извршуваат програмите туку за областа од која е проблемот што се решава и се нарекуваат **проблемско ориентирани јазици.**