

Omul și-a petrecut primul milion de ani ai existenței inventând și creând diverse unelte și mașini care i-au permis să se miște mai repede și să realizeze mai mult. Se poate spune că în acest fel, omul a reușit să-și mărească puterea fizică de mii de ori. Cu toate acestea, ceea ce îl deosebește pe om de alte specii sunt uneltele/instrumentele pe care le-a dezvoltat pentru a extinde și a spori puterea intelectului său.

Primul instrument cu care omul și-a îmbunătățit abilitățile intelectuale a fost limba, adică vorbirea și scrisul. Vorbirea a permis unei persoane să transmită cunoștințe și să-și articuleze intelectul, iar scrisul a oferit o perspectivă asupra cunoștințelor care au fost formate și dezvoltate de zeci de generații anterioare. Rândurile pe care le citiți chiar acum sunt un exemplu ideal al puterii cuvântului scris, care vă permite să aflați ceva ce nu știați înainte.

Cele mai noi din arsenalul de instrumente intelectuale sunt computerele - mașini care i-au permis omului să-și extindă puterea mentală până la limite neimaginabile. Cursul din fața voastră cu siguranță nu ar fi existat dacă nu ar fi existat computere. Prin urmare, primul subiect al acestui curs este înțelegerea lumii complexe a computerelor.

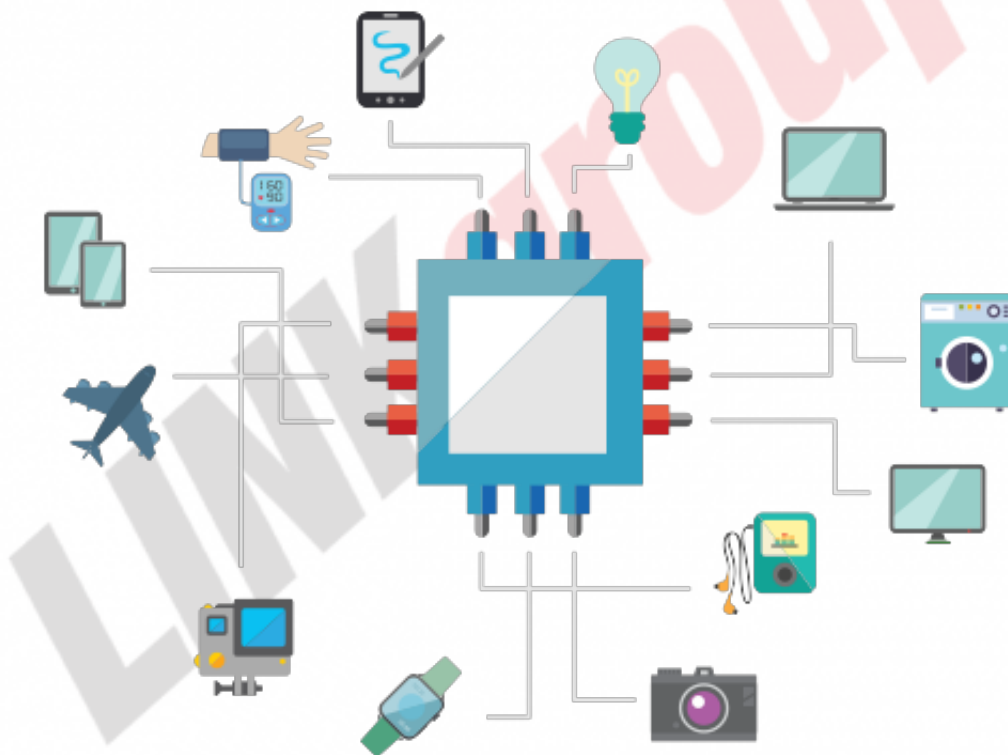
Înțelegerea lumii computerelor ne va ajuta semnificativ să urmărim în continuare acest curs, care va aborda dezvoltarea software-urilor sau, mai exact, postulatele designului și arhitecturii software.

## Lumea computerelor

Apariția computerelor a schimbat complet modul de viață al oamenilor. Astăzi, computerele sunt peste tot în jurul nostru, în [PC-uri și laptopuri](#), smartphone-uri, tablete, ceasuri inteligente, televizoare, mașini, avioane, mașini de spălat, aragaze, aparate de aer condiționat, bancomate...

Majoritatea activităților zilnice de astăzi nu pot fi imaginate fără un

computer. Deși poate nici nu vă dați seama, un număr mare de activități zilnice comune depind complet de computere. Activarea alarmei la un ceas cu alarmă digital, încălzirea micului dejun într-un cuptor cu microunde, călătoria cu mașina, retragerea banilor de la un bancomat, vizionarea filmului preferat la televizor - acestea sunt doar câteva exemple de activități de zi cu zi care nu ar fi posibile dacă nu ar exista computere. Deci, se poate spune pe bună dreptate că lumea de astăzi este lumea computerelor.



*Imaginea 1.1. Lumea computerelor*

Pe lângă activitățile umane de zi cu zi care sunt pe deplin susținute de computere, nici alte domenii ale lumii moderne de astăzi nu pot fi imaginate fără computere. Astfel, practic, nu există niciun domeniu al tehnologiei, artei sau științei care să folosească mai mult sau mai puțin computerele. Chirurgia cu laser a ochilor, proiectarea drumurilor, analiza probelor de sânge, tipărirea ziarelor, asamblarea mașinilor pe

linia de asamblare - acestea sunt doar câteva dintre operațiunile care astăzi nu pot fi imaginate fără computere.

## Ce este computerul?

O definiție comună a unui computer ar fi - *computerul este un dispozitiv capabil să efectueze un număr arbitrar de [operații aritmetice și logice](#)*. Dacă această propoziție este împărțită în mai multe părți, se poate spune că un computer este un dispozitiv care poate:

- accepta unele date de intrare,
- procesa acele date,
- produce rezultate care pot fi înțelese de un om sau de o altă mașinărie.

Cu alte cuvinte, un computer este o mașinărie care poate procesa date.

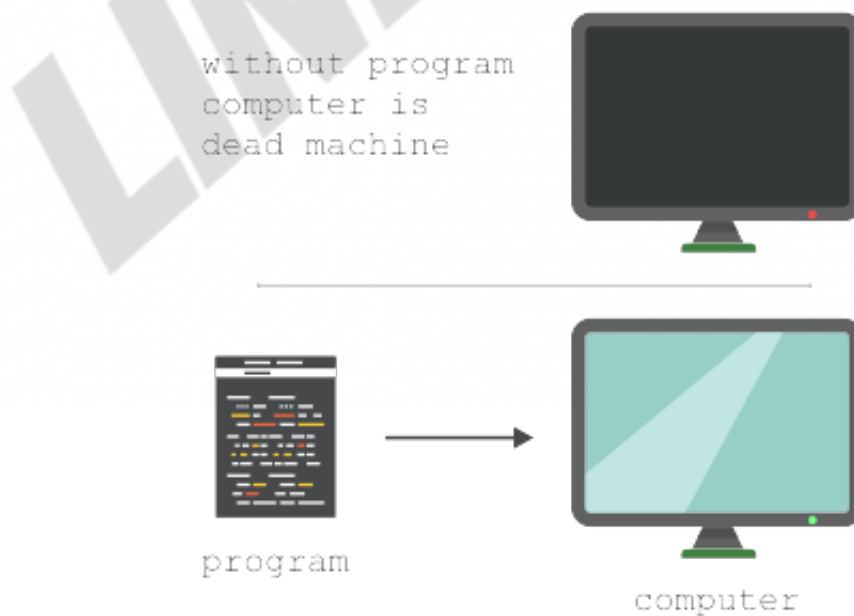
Deși termenul de computer este echivalat cu desktopurile și laptopurile, este foarte important să înțelegem că computerele vin într-o varietate foarte mare de forme. Mii de computere sunt ascunse în dispozitivele din jurul nostru, deși în unele situații nici nu suntem conștienți de asta. Astfel de computere pot fi numite specializate, deoarece sunt destinate să efectueze operații predeterminate. De exemplu, computerele din mașinile moderne sunt proiectate pentru a controla funcționarea motoarelor și a altor dispozitive. Astfel, computerul principal din mașină este destinat să controleze alimentarea cu combustibil, funcționarea supapelor și altele. Un astfel de computer, desigur, nu poate fi folosit pentru navigarea pe internet. Prin urmare, computerele care pot fi numite universale sunt mult mai aproape de noi.

Computerele universale sunt destinate să efectueze un set larg de operațiuni universale, care depind exclusiv de nevoile utilizatorului.

Computerele universale există astăzi în principal sub formă de computere desktop, laptop și tablete, precum și sub formă de smartphone-uri. Astfel de dispozitive pot fi folosite pentru a efectua o mare varietate de operațiuni, de la simpla navigare pe internet, scrierea textului, vizualizarea înregistrărilor audio și video, până la gestionarea și controlul altor tipuri de dispozitive digitale.

## Ce sunt programele?

Pe lângă mistica referitoare la computere în ansamblu, este important să înțelegem că acestea nu sunt mașinării magice, adică nu sunt capabile să facă nimic pe cont propriu. Fiecare computer este controlat de un program, care este, de fapt, un set de instrucțiuni care îi spun computerului ce trebuie să facă. Fără astfel de instrucțiuni, un computer ar fi o mașinărie care nu ar avea nicio utilizare practică (imaginea 1.2.).



*Imaginea 1.2. Programul dă viață computerului*

Cu setul potrivit de instrucțiuni, integrate în program, computerul poate deveni un instrument care poate fi folosit pentru comunicarea intercontinentală, lansarea unei rachete nucleare, cercetarea de noi molecule care pot vindeca cele mai grave boli etc. Având în vedere că aceste computere sunt create ca dispozitive fără nicio conștiință, este clar că procesul de scriere a instrucțiunilor, combinat într-un program, dă viață computerelor. Un astfel de proces se numește altfel programare.

## Ce este programarea?

Ceea ce diferențiază computerele de alte mașini, pe care omul le-a făcut de-a lungul timpului, este posibilitatea de programabilitate. Acest lucru înseamnă practic că un computer, în general, este un instrument universal care, în funcție de instrucțiunile care îi sunt transmise, poate îndeplini diverse sarcini. Tocmai în acest fapt este ascunsă cea mai mare putere a acestor mașinării - arbitrariul complet al programabilității.

Procesul de creare a programelor de calculator se numește **programare**. Astfel, programarea este actul de scriere a instrucțiunilor care permit unui computer să efectueze o operație. Programarea este un proces extrem de creativ care combină diverse aspecte ale artei, ingineriei și științei.

## Cine sunt programatorii?

Programele de calculator sunt create de programatori, oameni care au cunoștințele necesare pentru a putea scrie instrucțiuni care rulează computerul. Programarea modernă pe computere poate fi considerată o profesie relativ nouă, veche de doar puțin peste cincizeci de ani. Cu toate acestea, primele forme de programare au fost înregistrate mult mai devreme, în momentul existenței [computerelor mecanice](#), de o construcție complet diferită de cele pe care le cunoaștem astăzi.

Prima programatoare de computere din lume este considerată a fi **Ada**

**Lovelace** (imaginea 1.3.), fiica poetului Lord Byron.



*Imaginea 1.3. Ada Lovelace, prima programatoare din lume<sup>1</sup>*

În anul 1840, Ada Lovelace a primit la tradus un text profesionist al lui [Charles Babbage](#), pionierul construirii calculatoarelor. Textul conținea o prezentare a designului și funcționării noului său computer mecanic numit *Analytical Engine*. În timp ce traducea acel text din franceză în engleză, Ada Lovelace a scris primul set de instrucțiuni, al căror scop era să permită calcularea [numerelor Bernoulli](#) pe un computer mecanic. În cinstea acestei femei inovatoare, în 1980, Departamentul de Apărare al Statelor Unite a numit un nou limbaj pentru uz intern, Ada.

## **Limbajele de programare**

*Computerul, programul, programatorul și programarea* sunt termeni interdependenți. Un programator, printr-un proces numit programare,



crează un program care îi spune computerului ce trebuie să facă. Cu toate acestea, acestui proces rotunjit, pentru a fi complet, îi lipsește încă un element. Este vorba despre limbajul în care programatorul, în timpul procesului de creare a programului, i se adresează mașinii.

În viața de zi cu zi, oamenii comunică între ei folosind limbi naturale (*natural languages* în engleză), cum ar fi engleza, rusa, româna etc. Limbajul nu este altceva decât un set predeterminat de reguli, care sunt aplicate pentru a articula și prezenta gândurile altei persoane. Procesul de comunicare cu computerele se desfășoară într-un mod identic. Cu toate acestea, computerele nu înțeleg niciuna dintre limbile naturale pe care le folosesc oamenii, așa că trebuie să se adreseze computerului în limba lor maternă. Un astfel de limbaj este altfel numit limbaj mașină.

## Limbajul mașină

Limbajul mașină este limba maternă a computerelor, adică singura limbă pe care o înțeleg computerele. Codul de program al unui astfel de limbaj este alcătuit dintr-o serie de zerouri și unu, iar fiecare dintre aceste cifre se numește **bit**.

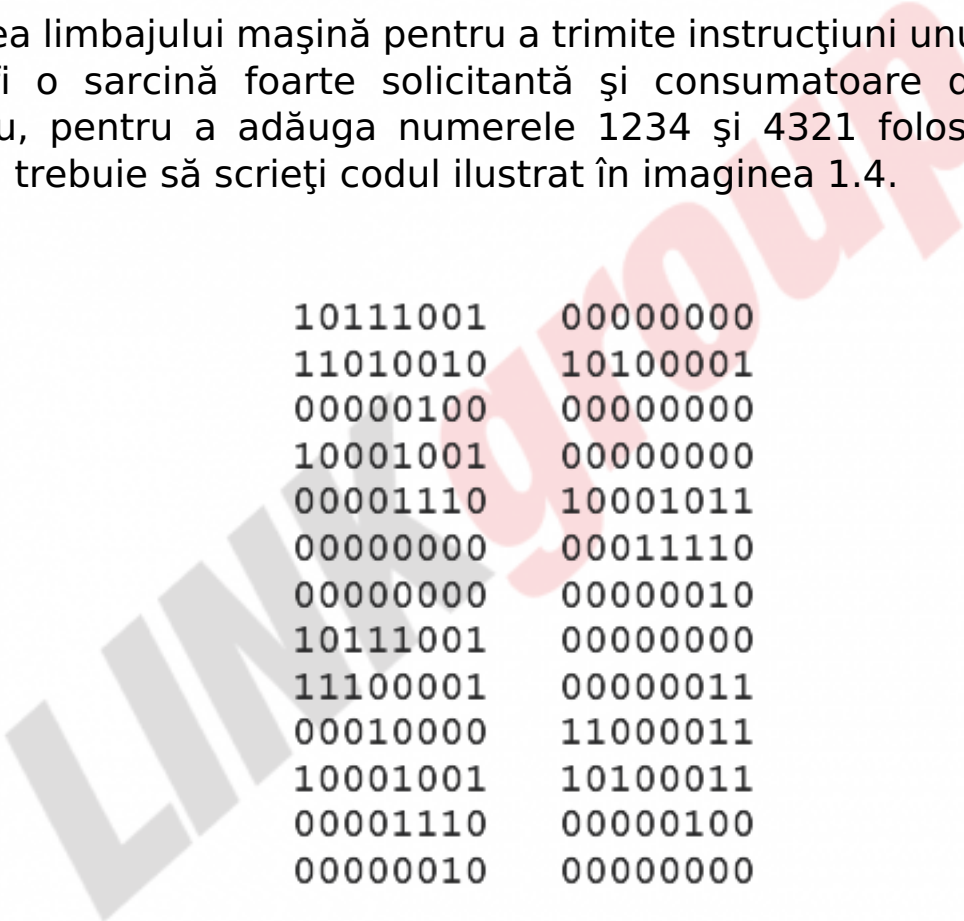
## Sistemul de numere binare

*Sistemul de numere binare este sistemul de bază pentru reprezentarea datelor într-un computer. Un procesor de computer gestionează doar date binare, în timp ce formatul identic este folosit pentru a stoca date în orice tip de memorie de computer.*

*Sistemul de numere binare, spre deosebire de sistemul de numere zecimale, are doar două cifre, 0 și 1. Utilizarea unui astfel de sistem de numere în electronica digitală este condiționată de proprietățile elementelor în sine, care sunt alcătuite din componente ale computerului, în primul rând din procesoare. Procesoarele moderne sunt compuse din sute de milioane de elemente semiconductoare (tranzistoare), care acționează ca întrerupătoare, care pot avea două*

*stări (pornit și oprit, adevărat și fals, zero și unu). Tocmai din cauza unor astfel de caracteristici structurale ale elementelor computerului, electronica digitală se bazează pe sistemul de numere binare.*

Utilizarea limbajului mașină pentru a trimite instrucțiuni unui computer, poate fi o sarcină foarte solicitantă și consumatoare de timp. De exemplu, pentru a adăuga numerele 1234 și 4321 folosind limbajul mașină, trebuie să scrieți codul ilustrat în imaginea 1.4.



10111001	00000000
11010010	10100001
00000100	00000000
10001001	00000000
00001110	10001011
00000000	00011110
00000000	00000010
10111001	00000000
11100001	00000011
00010000	11000011
10001001	10100011
00001110	00000100
00000010	00000000

*Imaginea 1.4. Programul pentru adunarea numerelor 1234 și 4321 scris cu limbajul mașină*

Setul prezentat de comenzi scrise în limbajul mașină, reprezintă programarea la nivel inferior, adică adresarea directă către procesorul computerului. Seturile de biți prezentate, adică seturile de zerouri și unu, reprezintă toate instrucțiunile necesare care sunt trimise computerului pentru ca acesta să efectueze munca pe care și-a imaginat-o programatorul.



Exemplul tocmai prezentat a ilustrat codul mașină care efectuează adunarea a două numere. Acum imaginați-vă o situație în care ar trebui să scrieți un program puțin mai complex folosind limbajul mașină, sau chiar un program care s-ar fi executat pe computerele moderne. Pe bună dreptate se poate spune că o astfel de muncă l-ar obosi chiar și pe cel mai persistent programator. Prin urmare, primul pas în simplificarea muncii programatorului a avut loc în anul 1949, când diferitor instrucțiuni binare le-au fost atribuite nume mai ușor de înțeles, care reflectau scopul lor. În acest fel, a fost creat limbajul de asamblare sau, pe scurt, asamblatorul.

## Asamblatorul

Asamblatorul, adică limbajul de asamblare (*assembler*, *assembly language* în engleză), adesea abreviat ca `asm`, este un limbaj de programare de nivel inferior, în care comenzile de bază ale programului sunt scrise folosind cuvinte predefinite care simbolizează comanda pe care o reprezintă. Astfel de construcții ale limbajului de asamblare sunt deseori numite simboluri, iar asamblatorul însuși este numit limbaj simbolic. Prin folosirea denumirilor pentru scrierea comenzilor programelor, limbajul de asamblare a făcut mult mai confortabilă scrierea programelor, având în vedere că prin folosirea asamblatorului, programatorul nu trebuia să-și amintească seria de zerouri și unu, din care erau făcute comenzile. Cu toate acestea, în ciuda faptului că este mult mai simplu de înțeles, limbajul de asamblare funcționează identic cu limbajul mașină, deoarece există aproape întotdeauna o relație unu la unu între instrucțiunile de asamblare și mașină.

Sarcina de adunare a numerelor 1234 și 4321, a căror performanță a fost ilustrată anterior folosind limbajul mașină, de data aceasta folosind limbajul de asamblare, poate fi realizată ca în imaginea 1.5.

```
MOV CX,1234  
MOV DS:[0],CX
```

```
MOV CX,4321  
MOV DS:[2],CX
```

```
MOV AX,DS:[0]  
MOV BX,DS:[2]
```

```
ADD AX,BX
```

```
MOV DS:[4],AX
```

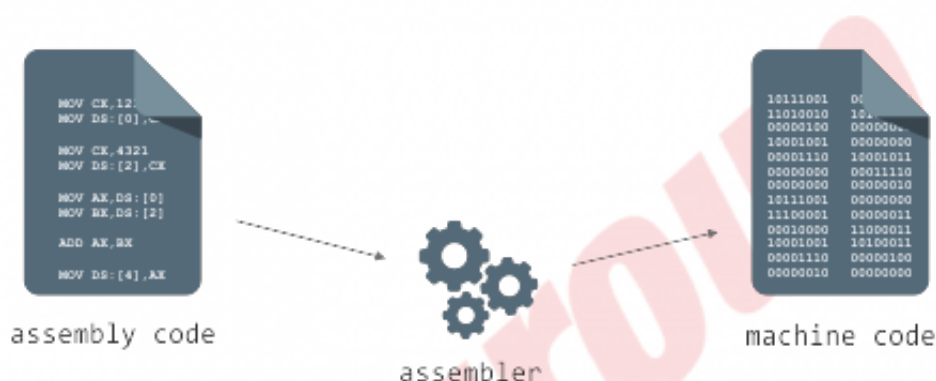
*Imaginea 1.5. Programul pentru adunarea numerelor 1234 și 4321 scris cu limbajul de asamblare*

După cum puteți vedea, codul de programare în limbajul de asamblare este mult mai ușor de înțeles decât codul mașină. În loc de seturi de zerouri și unu, comenzile sunt acum scrise în cuvinte englezești. De exemplu, `MOV` înseamnă mutare, adică scriere, iar `ADD` înseamnă adăugare. `AX`, `BX`, `CX` și altele reprezintă locații de memorie, adică registrele procesorului utilizate pentru a citi și scrie date. După cum puteți trage concluzia, apariția limbajului de asamblare a fost un mare pas înainte pentru profesia de programator.

Cu ceva timp în urmă am spus că computerele, adică unitățile de procesare, pot înțelege doar limbajul mașină. În rândurile anterioare, am vorbit despre asamblator, așadar, întrebarea care se pune este următoarea - *cum a învățat computerul un nou limbaj peste noapte și cum a devenit capabil să înțeleagă codul de asamblare pe lângă codul mașinii?*

Desigur, computerul nu a învățat instrucțiunile asamblatorului, ci acestea sunt compilate sub formă de cod mașină, astfel încât

procesorul să le poată înțelege înainte de execuție. Fiecare instrucțiune de asamblare este compilată într-un set corespunzător de zerouri și unu. Acest lucru este realizat de un program numit **asamblator (assembler)**. El este responsabil de compilarea instrucțiunilor de asamblare în codul mașinii (imaginea 1.6.).



*Imaginea 1.6. Compilarea codului de asamblare în codul mașină*

Din imaginea 1.6. puteți vedea cum un program de asamblare convertește instrucțiunile scrise în limbajul de asamblare în codul mașină. În mod clar, magia completă a limbajului de asamblare este localizată într-un software numit asamblator.

La fel ca limbajul mașină, limbajul de asamblare depinde în întregime de setul de instrucțiuni care au suport pe procesor. Aceasta înseamnă că procesoarele cu arhitecturi de instrucțiuni diferite necesită programe de asamblare diferite.

Apariția asamblatorului a facilitat foarte mult munca programatorilor. Totuși, acest lucru nu a fost suficient să simplifice, accelereze și standardizeze și mai mult munca de scriere a programelor de calculator. Prin urmare, abstracția a fost ridicată cu un nivel, prin crearea limbajelor de programare de nivel superior.

## Limbajele de nivel superior

Ambele tipuri de limbaje de programare cărora le sunt dedicate rândurile anterioare sunt considerate limbaje de nivel inferior. Limbajul mașină și limbajul de asamblare depind complet de caracteristicile fizice ale procesorului pe care sunt executate și din partea programatorului necesită multe cunoștințe despre arhitectura fizică a computerului în sine. Prin urmare, limbajele de nivel inferior sunt foarte dificil de învățat și necesită mult timp pentru a scrie programe. Tocmai din această cauză, de-a lungul timpului a fost creat un număr mare de limbaje de programare de nivel superior, care, prin abstracția detaliilor fizice ale computerului, au simplificat și accelerat semnificativ munca de scriere a programelor.

Spre deosebire de limbajele de nivel inferior, limbajele de programare de nivel superior folosesc într-o măsură mult mai mare elemente ale limbajelor naturale, în principal cuvinte și abrevieri preluate din limba engleză, îmbunătățind astfel lizibilitatea și facilitând crearea codului de programare. Pentru a înțelege cel mai bine avantajele unui limbaj de nivel superior, în imaginea 1.7. se poate vedea un cod de programare.

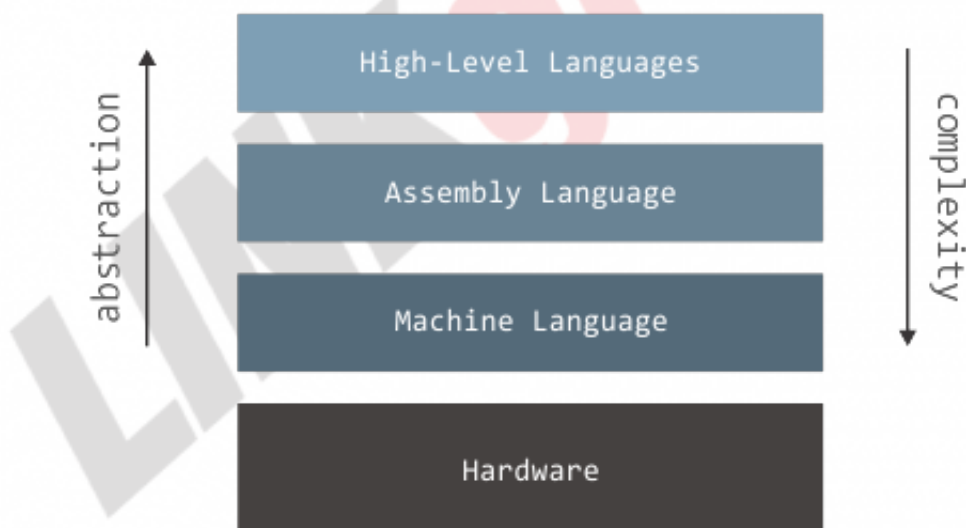
```
int a,b,c;  
  
a = 1234;  
b = 4321;  
  
c = a + b;
```

*Imaginea 1.7. Programul pentru adunarea numerelor 1234 și 4321 scris cu limbajul C*

Realizarea problemei identice, care a fost deja ilustrată folosind

limbajul mașină și de asamblare în exemplul din imaginea 1.7., se face folosind limbajul de programare C, unul dintre limbajele de nivel superior. Ceea ce puteți observa imediat este codul de programare semnificativ mai scurt, mai simplu și mai ușor de înțeles.

Pe lângă reducerea complexității, limbajele de nivel superior aduc și sisteme diferite care scutesc complet programatorul de nevoia de a se ocupa de anumite aspecte ale execuției programului. De exemplu, majoritatea limbajelor de programare de nivel superior gestionează automat memoria utilizată de un program în timpul execuției. Prin urmare, se poate spune că limbajele de nivel superior reduc complexitatea scrierii programelor prin introducerea unui grad mai ridicat de abstractizare. Imaginea 1.8. ilustrează acest lucru.



*Imaginea 1.8. Abstractizarea și complexitatea la diferite tipuri de limbaje de programare*

După cum se poate vedea în imaginea 1.8., prin trecerea de la hardware la limbajele de nivel superior crește abstracția și scade complexitatea. Acest lucru înseamnă practic că limbajele de nivel superior sunt caracterizate de un grad mai mare de abstractizare, iar limbajul mașină este caracterizat de o complexitate semnificativ mai

mare.

De-a lungul timpului, a fost creat un număr mare de limbaje de programare de nivel superior. Dacă începeți să învățați programarea astăzi, aproape sigur că veți învăța unul dintre limbajele de nivel superior. De asemenea, diferitele limbaje de nivel superior diferă foarte mult între ele în ceea ce privește capabilitățile, performanța, [sintaxa](#) și nivelul de abstractizare. Unele limbaje de nivel superior permit, de asemenea, implicarea în programarea la nivel inferior (low level), permițând ca părți ale unui program de calculator să fie scrise în limbajul de asamblare al unui anumit tip de procesor.

Programele scrise în limbajele de nivel superior trebuie compilate în limbajul mașină înainte ca un computer să le poată executa, așa cum a fost cazul codului de asamblare. Pentru a efectua un astfel de lucru, sunt utilizate două componente foarte importante ale programării moderne – compilatoarele și interpretorii. Într-una din lecțiile următoare, în care vom vorbi despre instrumentele de programare de bază, vom folosi ambele instrumente tocmai menționate pentru a compila codul sursă pe care îl scriem într-o formă care poate fi înțeleasă de computer.

## **Limbajele de programare moderne utilizate pe scară largă**

În rândurile anterioare au fost menționate unele dintre limbajele de programare de nivel superior care sunt utilizate astăzi pe scară largă. Fără intenția de a clasifica vreun limbaj în vreun fel, în tabelul 1.1. vă oferim o prezentare generală a celor mai populare limbaje de programare de astăzi.

<b>Limbaj de programare</b>	<b>Descriere</b>
C	Predecesorul majorității limbajelor moderne, C a influențat semnificativ majoritatea limbajelor de programare enumerate în acest



C++

tabel. Astăzi, este folosit predominant pentru programarea sistemelor de nivel inferior; de exemplu, baza sistemului de operare Linux (nucleul Linux) este scrisă în întregime folosind limbajul de programare C; de asemenea, o mare parte a sistemului de operare Windows este scrisă în acest limbaj. Creatorul limbajului de programare C este Dennis Ritchie. C++ este un alt limbaj al generației mai vechi care este încă folosit în timpurile moderne. A fost modelat după limbajul C, cu mai multe extensii pe care limbajul C nu le avea. Astăzi, este folosit în diverse domenii ale programării, de la programare de nivel inferior până la aplicații cu interfață grafică cu utilizatorul. Cu toate acestea, din cauza complexității, pentru programarea de nivel superior, programatorii folosesc cel mai adesea alte limbaje care sunt disponibile. A fost creat în anul 1983 de Bjarne Stroustrup.

C#

C# este unul dintre limbajele din familia .NET, creat de compania Microsoft. Este vorba de un limbaj de uz general, care este utilizat pe scară largă astăzi pentru a crea aplicații desktop, web și mobile; a fost prezentat în anul 2000.

PHP

PHP este unul dintre cele mai populare limbaje în prezent,

## Java

destinat în principal programării web. Se caracterizează printr-o sintaxă simplă, deci, aparține limbajelor care nu sunt greu de învățat. Este un exemplu ideal de creștere constantă a abstractizării programării computerelor, adică de o distanță tot mai mare de modul în care funcționează un computer, în condițiile în care implementările acestui limbaj sunt scrise folosind C și C++. Este un exemplu ideal de utilizare a limbajelor existente pentru a crea altele noi, de nivel superior. A fost creat în anul 1995, iar creatorul său este Rasmus Lerdorf.

Java este un limbaj de programare de uz general, deci, poate fi folosit pentru a scrie aplicații pentru o gamă foarte largă de dispozitive; care este și cel mai mare avantaj al acestui limbaj, care a fost creat cu scopul de a face posibilă execuția lui pe cât mai multe dispozitive computerizate diferite. Astăzi, însă, Java este folosit mai ales pentru programarea web și dezvoltarea aplicațiilor pentru dispozitive cu sistemul de operare Android. Java a fost creat în 1995, iar creatorul său este James Gosling și compania Sun Microsystems.

## JavaScript

Limbajul de bază al programării aplicațiilor web client, JavaScript este unul dintre elementele de bază ale web-ului modern. Este

## SQL

folosit pentru a face paginile web mai interactive și este, de asemenea, tehnologia de bază folosită pentru a crea jocuri web și animație. Este parte integrantă a tuturor browserelor web moderne. A apărut pentru prima dată în anul 1995, ca parte integrantă a [Netscape](#). Creatorul său este Brendan Eich.

Acesta este un limbaj al cărui singur scop este manipularea datelor stocate în baze de date. Astăzi, există un număr mare de implementări ale acestui limbaj, care au fost create de companii IT de top. Acesta este un limbaj care permite în primul rând citirea, scrierea, ștergerea și modificarea datelor, dar și crearea bazelor de date și a tabelor, precum și definirea unei logici simple, bazat pe limbaje de programare de uz general; limbajul a fost creat în 1974 de compania IBM.

## Objective C / Swift

Două limbaje utilizate pentru crearea aplicațiilor pentru dispozitivele Apple. Objective C este limbajul care a fost folosit inițial pentru programarea aplicațiilor pentru sistemele de operare OS X și iOS. Objective C a fost creat în 1984 de Tom Love și Brad Cox, extinzând limbajul C cu caracteristici orientate pe obiecte preluate din limbajul de programare Smalltalk. Limbajul de programare Swift a fost introdus

Python

de Apple în anul 2014 ca alternativă la limbajul Objective C; limbajul Swift se caracterizează prin utilizarea unor abordări mult mai moderne, precum și prin o sintaxă simplificată.

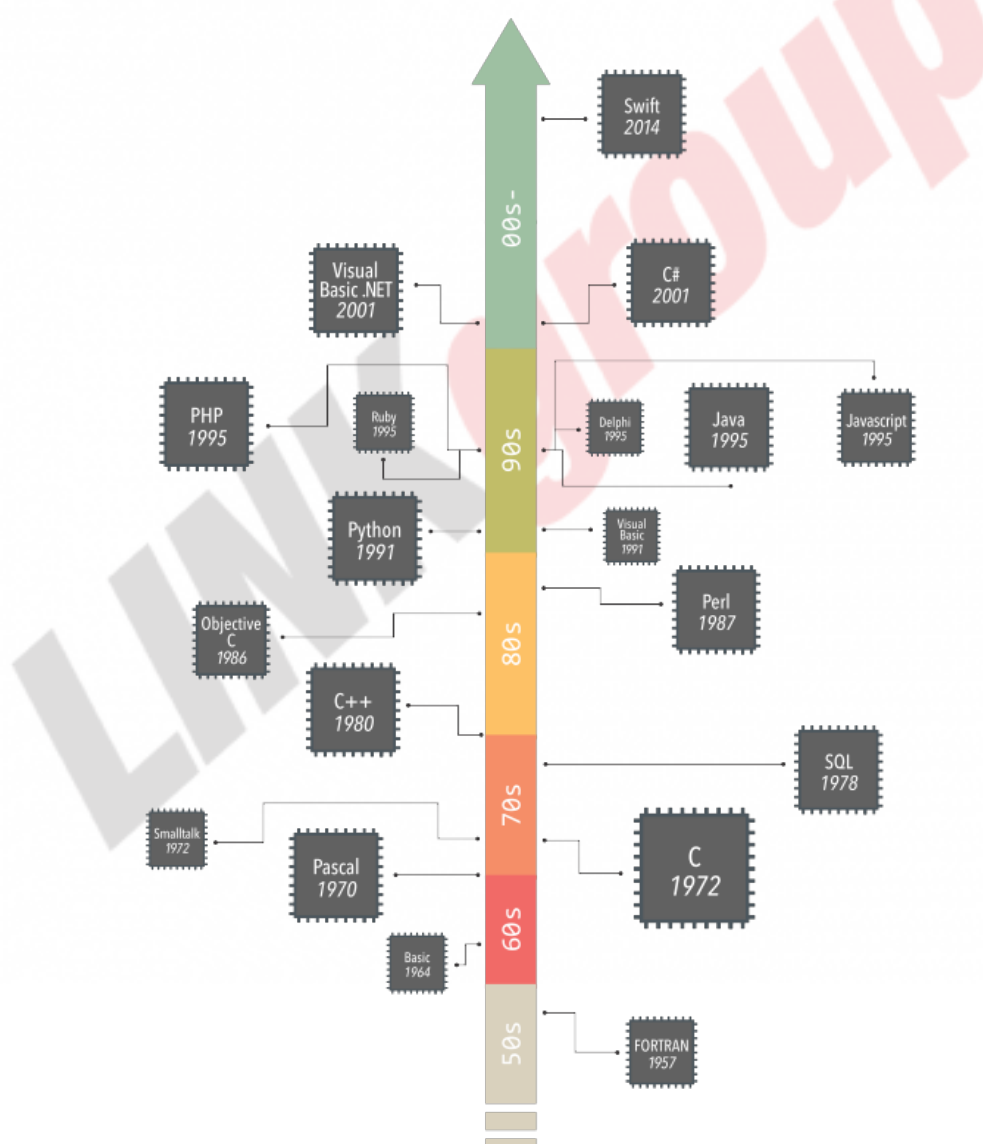
Python este un limbaj de programare de uz general ale cărui caracteristici principale sunt simplitatea sintaxei, lizibilitatea și concizia. Acest lucru înseamnă practic că prin utilizarea acestui limbaj, o problemă poate fi rezolvată cu mult mai puține linii de cod decât este cazul altor limbaje. Este vorba de un limbaj care este folosit astăzi în principal pentru programarea web, ca alternativă la PHP, Java sau C#. Este un exemplu clasic de limbaj al cărui cod sursă este interpretat. A apărut pentru prima dată în anul 1991, iar creatorul său este Guido van Rossum.

Perl

Încă un limbaj de uz general care poate fi folosit pentru a scrie diferite tipuri de software. Este unul dintre limbajele interpretate, iar interpretorul său poate fi integrat într-un număr mare de sisteme existente; este cel mai des utilizat pe web și este cunoscut în special pentru capacitățile sale bogate de procesare a textului; a fost creat în 1987 de Larry Wall.

*Tabelul 1.1. Cele mai populare limbaje de programare de nivel superior*

Pentru a recapitula tot ceea ce am spus în această lecție despre limbajele de programare, vă oferim imaginea 1.9., care ilustrează dezvoltarea lor de-a lungul timpului.



*Imaginea 1.9. Cronologia celor mai importante limbaje de programare de nivel superior*

## Începutul unei călătorii incitante

Citind această lecție, vă puteți imagina doar o parte din descoperirile interesante care se ascund în lumea dezvoltării software. Desigur, după această cantitate de informații, s-ar putea să vă simțiți confuz. Un astfel de sentiment este complet normal și caracteristic oricui intră în lumea labirintică a programării. Cu toate acestea, programarea este doar una dintre disciplinele care participă la procesul de dezvoltare software. Nu există niciun motiv de îngrijorare, având în vedere că acest program este conceput în așa fel încât să cunoașteți lumea dezvoltării software într-un mod sistematic și gradual. Călătoria incitantă poate începe.

---

1. (Sursa: [https://en.wikipedia.org/wiki/Ada\\_Lovelace](https://en.wikipedia.org/wiki/Ada_Lovelace))