

# Introduction to Software Development

<b>1. Lumea dezvoltării software</b> .....	2
1. Introducere în programare.....	2
2. Programele de calculator.....	2
3. Instrumente pentru dezvoltarea software-urilor.....	2
<b>2. Bazele programării</b> .....	4
4. Variabile și tipuri de date.....	4
5. Operatori.....	4
6. Controlul fluxului.....	4
<b>3. Inginerie software</b> .....	5
7. Noțiunea de inginerie software.....	5
8. Cerințele software.....	5
9. Testarea și mentenanța.....	6
<b>4. Instrumente pentru versiune, colaborare și managementul proiectelor</b> .....	7
10. Colaborare și versiune.....	7
11. Gestionarea proiectelor.....	7

# 1. Lumea dezvoltării software

## 1. Introducere în programare

- Computerul este un dispozitiv capabil să efectueze un număr arbitrar de operații aritmetice și logice.
- Programul este un set de instrucțiuni care îi spun computerului ce trebuie să facă.
- Procesul de creare a programelor de calculator se numește programare.
- Programele de calculator sunt create de programatori, oameni care au cunoștințele necesare pentru a putea crea instrucțiuni care direcționează modul în care funcționează computerul.
- Ada Lovelace este considerată a fi prima programatoare de computere din lume.
- Limbajul mașină este limba maternă a computerelor, adică singura limbă pe care computerele o înțeleg.
- Asamblatorul (assembler) sau limbajul de asamblare, 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.
- Limbajele de programare de nivel superior folosesc elemente ale limbajelor naturale într-o măsură mult mai mare, în principal cuvinte și abrevieri preluate din limba engleză, îmbunătățind astfel lizibilitatea și facilitând crearea codului de programare.
- Compilatorul și interpretorul sunt programe de calculator al căror scop principal este generarea altor programe de calculator; acestea două stau la baza funcționării limbajului de nivel superior.

## 2. Programele de calculator

- Programele de calculator sunt seturi de instrucțiuni care îi spun computerului ce trebuie să facă.
- Toate programele dintr-un computer sunt denumite în mod colectiv software.
- Programele de calculator sunt create prin scrierea codului programului folosind un limbaj de programare.
- Există multe tipuri de programe pentru calculator, în funcție de scopul lor, domeniul de execuție, dreptul de autor și alte caracteristici.
- În funcție de scop, programele de calculator pot fi clasificate în programe de sistem, de aplicație și programe malițioase.
- Programele de sistem sunt necesare pentru funcționarea dispozitivului, pentru manipularea hardware-ului acestuia, care ca atare permit altor programe să ruleze fără probleme pe un anumit dispozitiv.
- Programele de aplicație sunt destinate utilizatorilor finali pentru a îndeplini diverse sarcini, divertisment, comunicare, muncă...
- Programele malițioase sunt programe ale căror scop principal este de a provoca deteriorarea computerului și a datelor sau de a efectua o anumită operațiune care este contrară dorințelor utilizatorului.
- În funcție de domeniul de execuție, programele pot fi împărțite în programe desktop, web și mobile.
- Aplicațiile desktop sunt acele aplicații care sunt destinate computerelor desktop și laptop.
- Aplicațiile web necesită programe speciale numite browsere web pentru a fi utilizate și logica lor este executată în două locuri: pe client, adică în browserul web și pe server.
- Aplicațiile mobile sunt acele aplicații care rulează pe telefoane mobile, tablete și ceasuri inteligente.
- Programele de calculator pot fi clasificate în funcție de drepturile și obligațiile utilizatorului, iar în acest caz am abordat termenii de free software, open-source, copylefted, non-copylefted, freeware și alte tipuri de programe.

## 3. Instrumente pentru dezvoltarea software-urilor

- Un număr mare de tipuri diferite de instrumente participă la procesul de creare a software-ului; cunoașterea acestora este crucială pentru înțelegerea oricărei etape a dezvoltării programului.

- Instrumentele pentru proiectarea mediului de utilizator fac posibilă convenirea asupra aspectului detaliat și modului de funcționare a mediului de utilizator înainte de dezvoltarea programului; în acest scop pot fi folosite programele Adobe Photoshop, Adobe Illustrator, Adobe XD, Miro, Figma, Sketch, Zeplin etc.
- Scrierea codului sursă al programului se poate face folosind un editor de text sau un mediu de dezvoltare.
- Codul sursă este compilat la nivelul inferior în codul mașină, folosind instrumente numite interpretori și compilatoare.
- Editoarele de text sunt programe care permit scrierea codului sursă, marcarea acestuia, formatarea, copierea, modificarea, căutarea etc.
- Unele dintre cele mai populare editoare de text sunt Notepad++, Sublime Text, Atom, Visual Studio Code etc.
- Mediile de dezvoltare integrate (Integrated Development Environments, IDE pe scurt) sunt programe de dezvoltare software care, pe lângă editoarele de text, au și alte instrumente de dezvoltare: compilatoare, linkere, depanatoare, instrumente de versionare etc.
- Mediile de dezvoltare populare sunt: Visual Studio, NetBeans, IntelliJ IDEA, Android Studio, PyCharm, Xcode etc.

## 2. Bazele programării

### 4. Variabile și tipuri de date

- Miezul fiecărui limbaj de programare este capacitatea procesării a datelor.
- În programe, datele apar în numeroase forme, adică sub formă de numere, text, caractere și imagini, videoclipuri etc.
- Datele pe care le folosește programul în timpul execuției sunt stocate în memoria de lucru (RAM).
- În memoria de lucru, datele sunt înscrise în anumite locații de memorie.
- Variabilele sunt simboluri prin care se marchează locațiile de memorie.
- Procesul de creare a unei variabile și plasarea unei valori în ea reprezintă acțiunea care este alcătuită din două părți: declarare și inițializare.
- Declararea variabilei reprezintă rezervarea spațiului de memorie unde ulterior va fi stocată valoarea.
- Inițializarea variabilei reprezintă plasarea variabilei în spațiul de memorie rezervat.
- Clasificarea valorilor care pot fi plasate în variabile se efectuează datorită tipurilor de date.
- Două sisteme de tip de bază ale limbajelor moderne de programare sunt sistemele de tip dinamic și static.
- Toate limbajele de programare dispun de tipurile de date de bază; acestea sunt numere, text, caractere și valori logice.

### 5. Operatori

- Operatorii reprezintă elemente lexicale speciale ale limbajului care permit efectuarea operațiilor asupra valorilor.
- Operatorii aritmetici sunt cei care efectuează operații aritmetice asupra valorilor.
- Datorită operatorilor de atribuire, variabilele se atribuie unei valori.
- Operatorul de atribuire de bază este caracterul egal (=).
- Operatorii de comparare se utilizează pentru a compara două sau mai multe valori.
- Operatorii logici se utilizează pentru a se determina legătura logică dintre variabile, respectiv valori.
- Fiecare operator poate fi unar, binar sau ternar, în funcție de faptul dacă necesită unu, doi sau trei operanzi.

### 6. Controlul fluxului

- Controlul fluxului permite influențarea liniarității execuției codului.
- Ramificarea permite execuția anumitor instrucțiuni numai dacă este îndeplinită o condiție.
- Instrucțiunea de bază de ramificare este if.
- Instrucțiunea if constă dintr-o condiție și un bloc de cod care va fi executat dacă condiția este îndeplinită.
- Instrucțiunea if...else este folosită pentru controlul fluxului, permițând definirea logicii care va fi executată atât dacă condiția este îndeplinită, cât și dacă nu este îndeplinită.
- Buclele, pe lângă ramificare, sunt elementele de bază pentru realizarea controlului fluxului.
- Buclele asigură execuția unui cod de mai multe ori.
- Buclele populare sunt for și while.

## 3. Inginerie software

### 7. Noțiunea de inginerie software

- Ingineria software este o disciplină de inginerie care se ocupă de toate aspectele dezvoltării produselor software de la primele etape ale creării specificațiilor software până la livrare și mentenanță.
- Ingineria software a apărut în anii '60 ca răspuns la criza software.
- Criza software (Software crisis în limba engleză) este un termen care descrie situația în care s-a aflat dezvoltarea profesională a software-ului la începutul anilor '60, când programele de calculator nu puteau ține pasul cu progresul hardware-ului.
- În timpul crizei software, dezvoltarea programelor a depășit bugetele și termenele limită, iar programele în sine erau de proastă calitate, neoptimizate, nesigure, provocând adesea defecțiuni catastrofale ale sistemului și chiar pierderi de vieți omenești.
- Caracteristicile pe care trebuie să le aibă fiecare produs software de calitate sunt: ușurința de mentenanță, fiabilitatea și securitatea, eficiența și acceptabilitatea.
- Ingineria software rezolvă criza software cu procese de dezvoltare, profesionalism, instrumente software, paradigme, metodologii și metode formale.
- Procesul de dezvoltare software este o abordare sistematică care descrie o serie de activități ce se desfășoară în timpul dezvoltării unui produs software.
- Există patru activități de bază care alcătuiesc orice proces de dezvoltare software: specificarea software-ului, dezvoltarea software-ului, validarea software-ului și evoluția software-ului.
- reprezentarea simplificată a procesului software este altfel numită model de dezvoltare software.
- Modelul de dezvoltare în cascadă (waterfall) presupune împărțirea procesului de dezvoltare software într-un număr de etape interdependente, în care este necesară parcurgerea etapei precedente pentru a trece la următoarea.
- Modelul de dezvoltare agilă presupune împărțirea procesului de dezvoltare software într-un număr mare de unități mai mici, care se numesc incremente; dezvoltarea are loc prin iterații, unde în timpul unei iterații sunt parcurse toate etapele de dezvoltare, de la planificare, realizare până la testare, iar fiecare iterație are ca rezultat un software complet funcțional.

### 8. Cerințele software

- Specificația software este etapa de dezvoltare a software-ului care se ocupă de cerințele software.
- Cerințele software sunt o descriere a sistemului care trebuie construit cu toate caracteristicile și limitările așteptate.
- Utilizatorii finali, clienții, finanțatorii, inginerii care participă la dezvoltare - acestea sunt doar câteva exemple de entități care pot participa la crearea cerințelor.
- Fiecare cerință software ar trebui să fie: corectă, lipsită de ambiguitate, completă, consecventă, verificabilă, necesară, independentă și realizabilă.
- Cele două grupuri de bază de cerințe software sunt cerințele funcționale și nefuncționale.
- Etapele de lucru cu cerințele software sunt: colectarea cerințelor, analiza cerințelor, specificația cerințelor și validarea cerințelor.
- Cerințele pot fi colectate prin interviu, observație, chestionar, analiza sistemelor existente, sarcină temporară...
- Analiza cerințelor presupune înțelegerea cerințelor colectate.
- Specificația cerințelor este un proces în care cerințele software sunt notate într-o formă ceva mai formală, rezultând un așa-numit document SRS.
- Validarea cerințelor este procesul prin care se stabilește dacă cerințele definesc un sistem cu caracteristici de care au nevoie utilizatorii.
- Diagrama de context este folosită pentru a reprezenta limitele sistemului modelat și relația acestuia cu mediul, respectiv cu actorii.

- Diagrama fluxului de date este utilizată pentru a reprezenta modul în care datele se deplasează prin sistem.
- Diagramele entitate-relație reprezintă entități și atributele acestora, precum și relațiile dintre astfel de entități.
- Unified Modelling Language (UML) este un limbaj de modelare vizuală utilizat pentru a specifica, vizualiza, construi și documenta sisteme software.
- Diagrama de caz de utilizare ilustrează interacțiunea utilizatorului cu sistemul.
- Diagrama de activitate ilustrează fluxul de activități atunci când se efectuează o operațiune a sistemului.

## 9. Testarea și mentenanța

- Testarea software presupune verificarea programului, pentru a stabili că programul nu are erori software.
- Există un număr mare de tipuri diferite de teste software și de criterii pe baza cărora este posibilă clasificarea acestora.
- Testarea manuală presupune testarea în timpul căreia nu se utilizează un instrument de automatizare, adică un program special conceput pentru efectuarea testelor.
- Testarea automată implică efectuarea testelor folosind un instrument special, adică un program scris pentru a testa alte programe.
- Testarea Black Box este o tehnică care presupune testarea fără nicio cunoaștere a structurii interne a sistemului.
- Testarea White Box presupune cunoașterea completă a funcționării interne a sistemului.
- Testele unitare sunt concentrate pe funcționalitățile individuale care alcătuiesc programul complet.
- Testele de integrare sunt destinate testării interacțiunii dintre diferite componente mai mici ale unui sistem software.
- Testele de sistem au ca scop testarea întregului sistem în ansamblu, cel mai adesea din punctul de vedere al utilizatorului.
- Mentenanța software este o etapă a dezvoltării software care presupune actualizarea produsului software astfel încât acesta să răspundă în mod continuu așteptărilor utilizatorilor.

## 4. Instrumente pentru versiune, colaborare și managementul proiectelor

### 10. Colaborare și versiune

- Pentru a se organiza munca programatorului și facilita întreținerea, pe parcurs au fost create numeroase sisteme privind versiunea codului.
- Sistemele privind versiunea se mai numesc și version control, revision control și source control systems.
- Versiunea codului este o abordare care permite monitorizarea modificărilor codului sursă.
- Două dintre cele mai cunoscute sisteme privind versiunea sunt Git și SVN.
- Creatorul Git este Linus Torvalds, care a creat sistemul care, datorită simplității, vitezei și faptului că este un software open source, de-a lungul timpului a câștigat popularitate mare.
- Git este un instrument fără mediu de utilizator, astfel că utilizarea lui implică utilizarea unei console sau a unui terminal, în funcție de sistemul de operare care se utilizează.
- Comanda pentru verificarea existenței sistemului Git și a versiunii este git --version.
- Depozitul reprezintă figura centrală a sistemului Git; adesea este numit proiect Git și reprezintă o colecție completă a fișierelor din care este alcătuit un proiect software.
- Crearea depozitului Git se efectuează prin emiterea comenzii git init.
- Plasarea fișierelor și folderelor în depozit se efectuează prin întreprinderea a doi pași numiți staging și committing.
- Staging poate fi un pas pregătitor, iar committing acțiunea care inițiază înscrierea modificărilor în depozit.
- Staging-ul, adică pregătirea pentru livrare, se întreprinde folosind comanda git add.
- Commit, respectiv livrarea sau finalizarea, se întreprinde folosind comanda git commit.
- Verificarea stării fișierelor în depozitul Git se întreprinde folosind comanda git status.
- O privire de ansamblu asupra tuturor acțiunilor commit în depozit se poate întreprinde folosind comanda git log.
- GitHub este cel mai popular sistem online pentru găzduirea/hosting-ul proiectelor Git; alte sisteme cu scop identic sunt Bitbucket, GitLab, SourceForge etc.

### 11. Gestionarea proiectelor

- Fiecare proiect, indiferent de domeniul de activitate, este de fapt un grup de sarcini care trebuie implementate pentru a se obține rezultatul așteptat.
- Managementul proiectelor software (software project management, engleză) este disciplina care presupune planificarea și monitorizarea procesului de dezvoltare a unui produs software.
- Persoana care gestionează un proiect de dezvoltare software este de altfel numită manager de proiect.
- Managerul de proiect deține numeroase responsabilități: planificare, conducere, execuție, coordonarea timpului, coordonarea, întreținerea...
- Instrumentele de management de proiect sunt programe speciale de computer care sunt concepute pentru a ajuta echipele de dezvoltare în procesul de planificare, urmărire, colaborare și gestionare a proiectelor.
- Jira Software este un program pentru gestionarea proiectelor și urmărirea erorilor în timpul dezvoltării software.
- Jira Software este un produs comercial care a fost creat de compania Atlassian în 2002
- Există un număr mare de alternative la platforma Jira: Basecamp, Asana, Trello, Wrike...
- Unitatea organizatorică de bază a platformei Jira sunt proiectele.
- Crearea unui proiect presupune selectarea unui șablon și a unui tip de proiect.
- Trei modele de bază ale proiectelor Jira sunt Kanban, Scrum și Bug tracking.
- Jira permite crearea de proiecte care sunt administrate de echipa de dezvoltare însăși, dar și de cineva la nivel de companie.
- Locul central pentru organizarea tuturor articolelor de proiect ale platformei Jira este Jira Board.

- Proiectul Jira poate conține mai multe elemente care ilustrează sarcinile planificate, active și finalizate ale proiectului.