

**UNIVERSITATEA “Vasile Alecsandri”  
BACĂU  
FACULTATEA DE INGINERIE**

**CORNELIA NOVAC UDUDEC**

**INGINERIA SISTEMELOR DE  
PROGRAME**

*Ingineria programării*

*Ediție adăugită și revizuită*

**EDITURA ALMA MATER BACĂU  
2011**

## CUPRINS

<b>Capitolul 1 – Sisteme informatice. Probleme și perspective</b>	<b>Pag.</b>
1.1. Introducere	5
1.2. Probleme ale software-ului	11
1.3. Satisfacerea cerințelor utilizatorului și costul software	13
1.4. Performanța, portabilitatea și mentenanța software	16
1.5. Fiabilitatea	18
1.6. Cerințe pentru ingineria sistemelor de programe	18
1.7. Teorema pentru ingineria software	19
1.8. Clasificarea sistemelor de programe	25
1.9. Documentația sistemelor de programe	30
1.10. Dreptul de proprietate și garanții	33
1.11. Exerciții propuse	35
<b>Capitolul 2 – Etapele de dezvoltare a sistemelor de programe</b>	<b>37</b>
2.1. Ciclul de viață	37
2.2. Cerințe- Specificații	46
2.3. Concepte ale specificațiilor de programe	52
2.4. Specificarea formală	57
2.5. Exemple de specificare formală	59
2.6. Exerciții	66
<b>Capitolul 3 – Paradigmele de dezvoltare a sistemelor software</b>	<b>69</b>
3.1. Etapele de dezvoltare software	69
3.2. Paradigmele de dezvoltare software	71

- Metodologia cascadă	78
- Metodologia spirală	81
- Metodologia spirală WinWin	83
- Prototipizarea	84
- Metode formale	88
- Metoda V	89
- Programarea extremă	89
- Metoda Open Source	93
- Reverse Engineering	94
- Metoda de dezvoltare Offshore	94
- Metodologia orientată pe obiect	95
<b>Capitolul 4 – UML- Limbaj unificat de modelare</b>	<b>101</b>
4.1. Introducere în UML	101
4.2. Diagrame și concepte UML	104
- Diagrama claselor	105
- Diagrama cazurilor de utilizare	112
- Diagrama de stare	115
- Diagrama de activitate	119
- Diagrama secvențiale	122
- Diagrama de colaborare	125
- Diagrama de aplicație	129
<b>Capitolul 5 – Principii de proiectare orientată pe obiect</b>	<b>133</b>
5.1. Principiul Deschis-Închis	133
5.2. Principiul substituției Liskov	137
5.3. Principiul Inversării Dependențelor	144
5.4. Stabilitate. Principiul dependențelor stabile	151
<b>Capitolul 6 – Șabloane de proiectare software</b>	<b>161</b>
6.1. Elementele unui șablon de proiectare	161
6.2. Cum rezolvă șabloanele problemele de proiectare	166
6.3. Cum se selectează un șablon	167
6.4. Cum se folosește un șablon de proiectare	169

<b>Capitolul 7 – Proiectarea sistemelor software</b>	173
7.1. Procesul de proiectare	173
7.2. Proiectarea arhitecturală	177
7.3. Proiectarea calității	180
7.4. Modularizarea proiectului	186
<b>Capitolul 8 – Testarea sistemelor software</b>	191
8.1. Introducere	191
8.2. Testarea pe parcursul ciclului de viață al unui program	192
8.3. Testele de sistem	196
8.4. Determinarea cazurilor de test	197
<b>Capitolul 9 – Estimarea costurilor unui proiect software</b>	211
9.1. Costuri și efort	211
9.2. Modelul Halstead	212
9.3. Modele algoritmice clasice – Modele liniare	214
- Modelul Nelson	214
- Modelul Wolverton	215
9.4. Modele algoritmice moderne – Modele neliniare	215
- Modelul Walston – Felix	216
- Modelul COCOMO	217
- Modelul Putnam – Norden	221
- Legea lui Brooks	223
- Mărimea echipei și productivitatea	223
<b>Capitolul 10 – Calitatea sistemelor software</b>	225
10.1. Indicatori de calitate	225
10.2. Productivitatea	228
10.3. Asigurarea fiabilității produselor software	229
10.4. Metrice software pentru paradigma orientată obiect	233
10.5. Modele de studiere „a posteriori” a fiabilității	237
10.6. Modele software pentru reducerea erorilor	247
10.7. Utilizarea datelor eronate pentru îmbunătățirea deciziilor	252
10.8. Reducerea erorilor datorită măsurătorilor	254

10.9. Aplicarea analizei cauzale procesului de modificare a software-ului	256
<b>Capitolul 11 – Evaluarea sistemelor software</b>	<b>267</b>
11.1. Set de metrici software pentru conducerea proceselor de mentenanță a programelor	267
- Definirea setului de metrici	268
11.2. Program de implementare a metricilor	281
<b>Bibliografie</b>	<b>283</b>