

# **Bazele programării I**

## **Metode de proiectare a algoritmilor**

# Programarea structurată

Un **program structurat** este format din unități funcționale bine conturate, ierarhizate conform naturii problemei, cu o structură bine precizată, atât la nivelul instrucțiunilor, cât și al datelor, obținută printr-un proces de rafinare treptată.

**Principiile** de bază ale programării structurate:

- proiectarea descendentă (de sus în jos);
- orice algoritm se poate construi folosind doar patru tipuri de structuri de control;
- domeniile de vizibilitate ale variabilelor și structurilor de date trebuie să fie limitate.

# Avantajele programării structurate

- structurarea determină o definiție exactă a funcțiilor fiecărui modul;
- scrierea programului și punerea lui la punct se face mai simplu și mai rapid;
- modificarea și extinderea programului se poate face ușor, prin modificarea anumitor module și adăugarea unor module noi;
- fiecare modul se poate scrie în limbajul de programare cel mai avantajos în raport cu funcția sa.

# Proiectarea algoritmilor prin metoda "de sus în jos"

1. Divizarea problemei inițiale în subprobleme mai simple.
2. Subproblemele complicate se divid iarăși în subprobleme mai simple.
3. Procesul de divizare a subproblemelor se termină atunci când în rezultatul ultimei proceduri s-au obținut subprobleme simple.
4. Soluția problemei inițiale se obține prin reuniunea soluțiilor subproblemelor.

Procesul de divizare a problemelor complicate în subprobleme mai puțin complicate se numește **detaliere**.

# Descompunerea problemei în subprobleme

introducerea  
datelor inițiale;

calcularea  
datelor-rezultat;

afișarea  
rezultatelor.

# Problema casierului

## Denumirea problemei

*Problema casierului.*

## Descrierea problemei

*Se imită lucrul casierului. De la tastatură se introduce costul cumpărăturii și suma de bani primită de la cumpărător. Casierul calculează restul, care trebuie să-l returneze cumpărătorului. Restul trebuie să fie format dintr-un număr minim de bancnote.*

## Introducerea datelor inițiale

*Se introduc 2 numere întregi (costul cumpărăturii și suma plătită), fiecare din rând nou.*

## Afișarea rezultatelor

*Se afișează restul calculat și numărul de bancnote de fiecare fel care vor fi returnate cumpărătorului.*

# Problema casierului

## Erori

- *dacă nu au fost introduse ambele numere, programul va aștepta următoarea introducere;*
- *dacă au fost introduse mai mult de două numere, atunci celelalte numere se ignorează;*
- *dacă în rând au fost introduse mai multe numere, atunci programul afișează un mesaj de eroare și se termină.*

## Exemplu

*Costul cumpărăturii 117*

*Suma plătită 200*

*Rest 83*

*Bancnote a câte 500 0*

*Bancnote a câte 200 0*

*Bancnote a câte 100 0*

*Bancnote a câte 50 1*

*Bancnote a câte 20 1*

*Bancnote a câte 10 1*

*Bancnote a câte 5 0*

*Bancnote a câte 1 3*

# Introducerea datelor inițiale

**1.1.** Introducerea costului cumpărăturii;

**1.2.** Introducerea sumei plătite.



# Calcularea rezultatelor

- 2.1.** calcularea restului;
- 2.2.** calcularea numărului de bancnote a câte 500 lei;
- 2.3.** calcularea numărului de bancnote a câte 200 lei;
- 2.4.** calcularea numărului de bancnote a câte 100 lei;
- 2.5.** calcularea numărului de bancnote a câte 50 lei;
- 2.6.** calcularea numărului de bancnote a câte 20 lei;
- 2.7.** calcularea numărului de bancnote a câte 10 lei;
- 2.8.** calcularea numărului de bancnote a câte 5 lei;
- 2.9.** calcularea numărului de bancnote a câte 1 leu.

# Afișarea rezultatelor

- 3.1.** afișarea restului;
- 3.2.** afișarea numărului de bancnote a câte 500 lei;
- 3.3.** afișarea numărului de bancnote a câte 200 lei;
- 3.4.** afișarea numărului de bancnote a câte 100 lei;
- 3.5.** afișarea numărului de bancnote a câte 50 lei;
- 3.6.** afișarea numărului de bancnote a câte 20 lei;
- 3.7.** afișarea numărului de bancnote a câte 10 lei;
- 3.8.** afișarea numărului de bancnote a câte 5 lei;
- 3.9.** afișarea numărului de bancnote a câte 1 leu.

# Detalierea subproblemei 1.1.

## Introducerea costului cumpărăturii.

### **Declararea variabilei:**

*Cost\_cump: Natural*

### **Algoritmul de rezolvare a subproblemei 1.1.**

*WriteString('Costul cumpărăturii?')*

*ReadNat(Cost\_cump)*

*WriteLn*

# Detalierea subproblemei 1.2.

## Introducerea sumei plătite.

### Declararea variabilei:

*Suma\_platita: Natural*

### Algoritmul de rezolvare a subproblemei 1.2.

*WriteString('Suma plătită?')*

*ReadNat(Suma\_platita)*

*WriteLn*

# Detalierea subproblemei 2.

## Calcularea rezultatelor

### **Declararea variabilelor:**

*Rest: Natural*

*B500: Natural*

*B200: Natural*

*B100: Natural*

*B50: Natural*

*B50: Natural*

*B20: Natural*

*B10: Natural*

*B5: Natural*

*B1: Natural*

# Detalierea subproblemei 2.

## Calcularea rezultatelor

### Calcularea restului:

$$Rest := Suma\_platita - Cost\_cump$$

### Calcularea numărul bancnotelor a câte 500 lei

$$B500 := Rest \div 500$$

### Calcularea numărul bancnotelor a câte 200 lei

$$X := Rest \bmod 500$$
$$B200 := X \div 200$$

### Calcularea numărul bancnotelor a câte 100 lei

$$X := X \bmod 200$$
$$B100 := X \div 100$$

### Calcularea numărul bancnotelor a câte 50 lei

$$X := X \bmod 100$$
$$B50 := X \div 50$$

# Detalierea subproblemei 2.

## Calcularea rezultatelor

### Calcularea numărul bancnotelor a câte 20 lei

$$X := X \bmod 50$$
$$B20 := X \div 20$$

### Calcularea numărul bancnotelor a câte 10 lei

$$X := X \bmod 20$$
$$B10 := X \div 10$$

### Calcularea numărul bancnotelor a câte 5 lei

$$X := X \bmod 10$$
$$B5 := X \div 5$$

### Calcularea numărul bancnotelor a câte 1 leu

$$B1 := X \bmod 5$$

# Detalierea subproblemei 3.

## Afişarea rezultatelor

### Afişarea restului

```
WriteString('Rest=')  
WriteNat(Rest)  
Writeln
```

### Afişarea bancnotelor

```
WriteString('Bancnote a câte 500 lei ')  
WriteNat(B500)  
Writeln  
*  
WriteString('Bancnote a câte 200 lei ')  
WriteNat(B200)  
Writeln  
*  
WriteString('Bancnote a câte 100 lei ')  
WriteNat(B100)  
Writeln
```



# Modelul descrierii algoritmului în pseudocod

**Algoritm** *<nume>*

**Var**

*<lista variabilelor utilizate>*

**Begin**

*<operații>*

**End**

# Descrierea algoritmului Casier

## Algoritmul Casier

**Var**

*Cost\_cump: Natural*  
*Suma\_platita: Natural*  
*Rest: Nnatural*  
*B500: Natural*  
*B200: Natural*  
*B100: Natural*  
*B50: Natural*  
*B50: Natural*  
*B20: Natural*  
*B10: Natural*  
*B5 : Natural*  
*B1 : Natural*  
*X: Natural*

**Lista variabilelor**

# Descrierea algoritmului Casier

**Begin**

```
WriteString('Costul cumpărăturii?')  
ReadNat(Cost_cump)  
Writeln  
WriteString('Suma plătită?')  
ReadNat(Suma_platita)  
Writeln
```

**Subproblema 1**

```
Rest:=Suma_platita - Cost_cump  
B500:= Rest div 500  
X:= Rest mod 500  
B200:=X div 200  
X:= X mod 200
```

**Subproblema 2**

# Descrierea algoritmului Casier

*B100:=X div 100*

*X:=X mod 100*

*B50:=X div 50*

*X:=X mod 50*

*B20:=X div 20*

*X:=X mod 20*

*B10:=X div 10*

*X:=X mod 10*

*B5:=X div 5*

*B1:=X mod 5*

**Subproblema 2**

# Descrierea algoritmului Casier

```
WriteString('Rest= ')
WriteNat(Rest)
Writeln
WriteString('Bancnote de 500 lei ')
WriteNat(B500)
Writeln
WriteString('Bancnote de 200 lei ')
WriteNat(B200)
Writeln
WriteString('Bancnote de 100 lei ')
WriteNat(B100)
Writeln
```

**Subproblema 3**

# Descrierea algoritmului Casier

```
WriteString('Bancnote de 50 lei ')
WriteNat(B50)
Writeln
WriteString('Bancnote de 20 lei ')
WriteNat(B20)
Writeln
WriteString('Bancnote de 10 lei ')
Writelnat(B10)
Writeln
WriteString('Bancnote de 5 lei ')
Writelnat(B5)
Writeln
WriteString('Bancnote de 1 leu ')
Writelnat(B1)
Writeln
```

**End**

**Subproblema 3**

# Sarcini pentru lucrul independent

1. Să se determine aria unui cerc în funcție de diametrul său.
2. Elaborați algoritmul care citește de la tastatură un număr natural, format din 4 cifre și determină suma primelor 2 cifre a acestui număr și suma ultimelor două cifre ale acestui număr.
3. Elaborați algoritmul care transformă viteza, exprimată în km/oră în m/s.