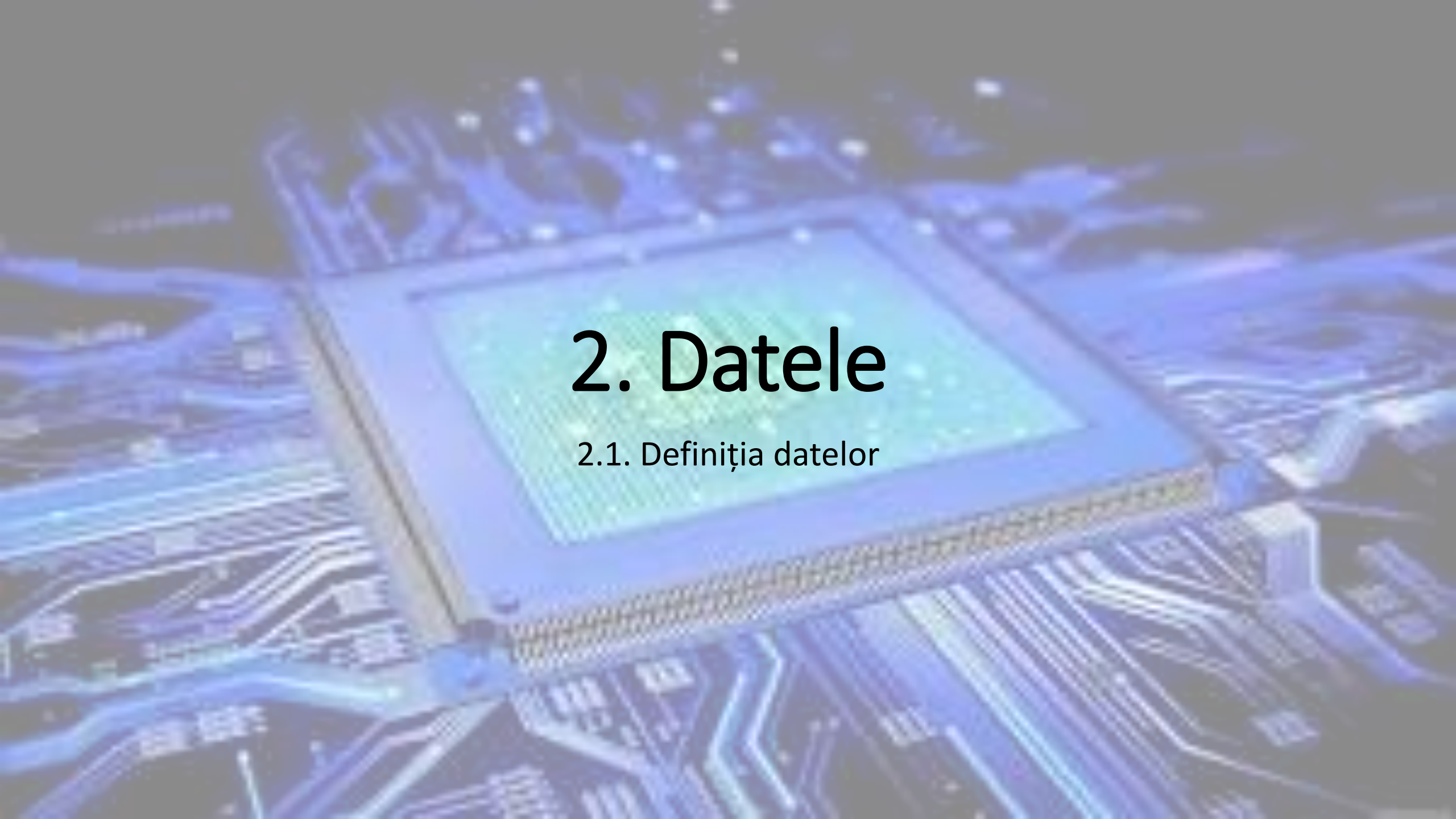


2. Datele

2.1. Definiția datelor

2.1. Definiția datelor

- Datele sunt obiectele prelucrate de algoritm

Data este un model de reprezentare a informaţiei, accesibil calculatorului, cu care se poate opera pentru a obţine noi informaţii.

- Din punct de vedere logic, data este definită printr-o tripletă:

data elementară = (identificator, valoare, atribute)

Identificatorul datei

- Identificatorul datei este un **nume** format din unul sau mai multe caractere şi este atribuit unei date de către cel care defineşte data, pentru a o putea distinge de alte date şi pentru a putea face referiri la ea în procesul de prelucrare a datelor. De exemplu: alfa, a11 sau b1_1.
- Fiecare limbaj de programare are implementat diferit conceptul de identificator al datei, practicând constrângeri pentru:
  - numărul maxim de caractere din nume (de exemplu, 10 caractere în cazul limbajului de programare Visual Basic şi 64 de caractere în cazul limbajului de programare Pascal);
  - caracterele acceptate în nume (de exemplu, majoritatea limbajelor de programare nu acceptă în nume decât cifre, litere şi caracterul linie de subliniere);
  - caracterul care poate fi folosit la începutul numelui (de exemplu, marea majoritate a limbajelor de programare nu acceptă ca numele unei date să înceapă cu o cifră).
- De aceea, atunci când învățați să definiți şi să manipulați date folosind un anumit limbaj de programare, trebuie să aflaţi ce constrângeri există pentru identificatorul datei.

Valoarea datei

- valoarea datei reprezintă conţinutul zonei de memorie în care este stocata data. Se defineşte ca domeniu de definiţie al datei, mulţimea valorilor pe care le poate lua data în procesul de prelucrare.

În zona de memorie alocată variabilei a, este păstrată valoarea 25.
În urma atribuirii a=17, conținutul zonei de memorie se actualizează cu noua valoare, cea veche pierzându-se.

Atributele datei

- Atributele sunt proprietăţi ale datelor care determină modul în care sistemul va trata datele. Cel mai important atribut este **tipul datei**.

Tipul datei defineşte apartenenţa datei la o anumită clasă de date, căreia îi corespunde un anumit model de reprezentare internă.

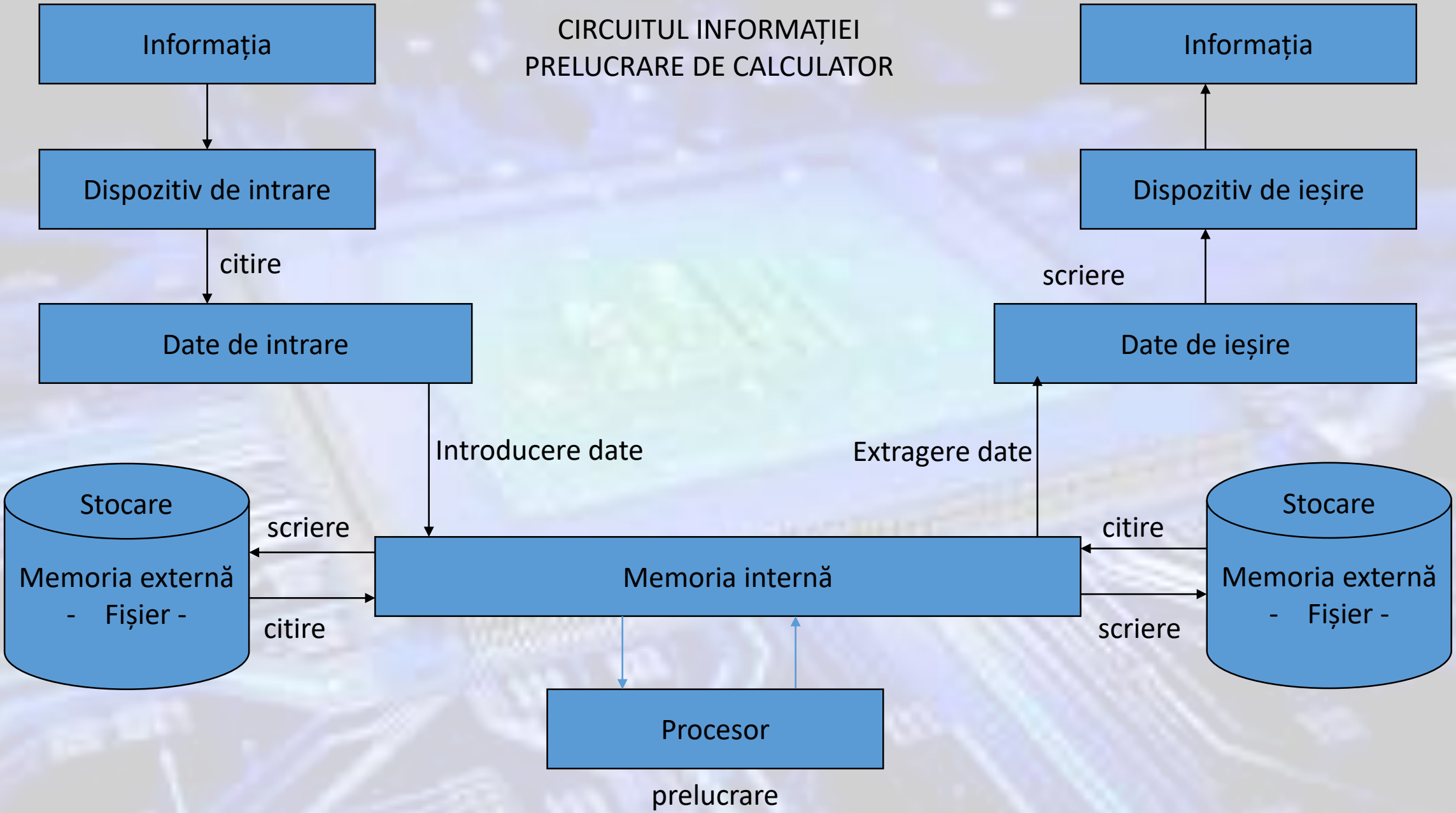
- Indiferent de tipul de date ales, reprezentarea datei în memoria calculatorului se face printr-un **şir de biţi**. Pentru a realiza această reprezentare, fiecare limbaj de programare are implementaţi **algoritmi de codificare** care asigură corespondenţa dintre tipul de dată şi şirul de biţi, atât la scrierea datelor, cât şi la citirea lor.
- Orice sistem care prelucrează informaţia sub formă de date trebuie să aibă definit clar conceptul de dată. Definirea conceptului de dată implică definirea următoarelor elemente:
  - cum poate fi identificată data?
  - cum va fi reprezentată data în memoria calculatorului?
  - ce proprietăţi are data?
  - cum pot fi grupate datele în colecţii de date?

2.1.1. Clasificarea datelor

- Clasificarea datelor se poate face folosind mai multe criterii:
  - în funcţie de momentul în care se produc în fluxul de informaţie:
    - date de intrare
    - date intermediare
    - date de ieşire
  - în funcţie de valoare:
    - date variabile
    - date constante
  - în funcţie de modul de compunere:
    - date elementare
    - structuri de date
  - în funcţie de tip.
    - date numerice
    - date logice
    - date şiruri de caractere

Clasificarea în funcţie de momentul în care se produc

- Datele se clasifică în:
  - **Date de intrare**. Ele reprezintă datele care urmează să fie prelucrate în cadrul algoritmului. Sunt folosite pentru a descrie diverse evenimente şi sunt informaţii produse în urma realizării unui eveniment, adică evaluări neprelucrate ale acelui eveniment. De exemplu, pot fi date de intrare notele unui elev. Pentru a putea fi prelucrate de procesor, datele sunt introduse în memoria internă a calculatorului. Introducerea dalelor se face prin intermediul unor echipamente specializate în citirea informaţiei, numite **dispozitive de intrare** (tastatură, scanner, creion optic etc). Dispozitivul standard de intrare este **tastatura**.
  - **Date de ieşire**. Ele sunt folosite pentru a descrie rezultatele obţinute în urma prelucrărilor din cadrul algoritmului şi furnizează informaţiile pentru care a fost realizat algoritmul, ca de exemplu mediile semestriale şi anuale ale elevului. Datele de ieşire sunt produse de procesor în urma operaţiei de prelucrare şi sunt depuse în memoria internă. Pentru a fi vizualizate de om, ele sunt extrase din memoria internă prin intermediul unor echipamente specializate în scrierea informației numite dispozitive de ieşire (ecran, imprimantă etc). Dispozitivul standard de ieşire este ecranul.
  - **Datele intermediare** sau de manevra. Ele sunt folosite, în cadrul algoritmului, pentru realizarea unor prelucrări.
- În vederea prelucrării, datele pot fi păstrate, temporar în memoria internă sau în memoria externă (discul flexibil, hard-discul, discul compact etc.). Operaţia se numește **stocarea datelor.**


CIRCUITUL INFORMAȚIEI
PRELUCRARE DE CALCULATOR
Informația
Dispozitiv de intrare
citire
Date de intrare
Introducere date
Informația
Dispozitiv de ieșire
scriere
Date de ieșire
Extragere date
Stocare
Memoria externă
- Fișier -
scriere
citire
Memoria internă
citire
scriere
Stocare
Memoria externă
- Fișier -
Procesor
prelucrare

- În memoria externă datele sunt păstrate în fişiere. Un fişier este o colecţie de date organizate ca o singură unitate. Dacă datele sunt păstrate în fişiere, ele vor putea fi folosite ulterior ca date de intrare într-un alt algoritm.
- Orice rezolvare de problemă începe prin **definirea datelor**, continuă cu **prelucrarea** lor în conformitate cu algoritmul folosit şi se termină fie cu **afişarea valorii** fie cu **stocarea** lor pe un mediu de memorare, în vederea prelucrării lor, ulterior.

Clasificarea datelor în funcţie de modul de compunere

- Dalele se clasifică în:
  - **Date simple** sau date elementare. Sunt date independente unele de altele din punct de vedere al reprezentării lor în memorie. Chiar dacă ele pot depinde din punct de vedere logic (valoarea unei date este dependentă de valoarea altei date), ele nu depind din punct de vedere fizic (localizarea unei date pe suportul de memorare nu se face în funcţie de locaţia pe suport a unei altei date).
  - **Date compuse** sau structuri de date. Sunt colecţii de date între care există anumite relaţii. Fiecare componentă a structurii are o anumită poziţie în cadrul structurii, iar toate componentele formează un întreg, astfel încât prelucrarea se poate face atât la nivelul structurii de date (care poale fi considerată o entitate de sine stătătoare), cât şi la nivelul fiecărei componente. Pentru fiecare tip de structură de dale, în limbajul de programare trebuie să fie definiţi algoritmi de localizare a componentelor în cadrul structurii de date. Între componentele structurii există şi legături de conţinut, adică întregul ansamblu de date din colecţie poate caracteriza un obiect, o persoană, un fenomen, un proces etc.

2.1.2. Tipul datei

- Tipul datei determină:
  - dimensiunea zonei de memorie alocată datei (se măsoară în octeţi);
  - operatorii care pot fi aplicați pe acea dată;
  - modul în care data este reprezentată în memoria internă (metoda de codificare în binar a valorii datei).
- Tipul datei este definit prin dubletul (V.O), unde
  - V = domeniul de definiție intern al datei;
  - O = mulțimea operatorilor care se pot aplica pe mulțimea de valori ale dalei.
- Limbajele de programare accepta următoarele tipuri de date.
  - tipul numeric
  - tipul logic
  - tipul şir de caractere

Tipul numeric

- **Tipul numeric** a fost implementat pentru reprezentarea numerelor întregi sau cu zecimale, pozitive sau negative, şi pentru a realiza majoritatea operaţiilor matematice întâlnite în practică
- Pentru tipul numeric există subtipurile **real** şi **întreg**.

V = $\mathbb{R}$ (mulţimea numerelor reale) sau $\mathbb{I}$ (mulţimea numerelor întregi)

O = $\mathcal{M} \cup \mathcal{R}$ ($\mathcal{M}$ este mulțimea operatorilor matematici și $\mathcal{R}$ este mulțimea operatorilor relaționali – de comparare)

- **Constantele de tip numeric** se reprezintă prin numere cu semn sau fără semn, folosindu-se punctul pentru separarea părţii întregi de partea zecimală: 2; -0.15; 3.175; 20.0.

Tipul logic

- Tipul logic, sau boolean, a fost implementat pentru reprezentarea datelor care nu pot lua decât două valori: adevărat (true), pe care o notăm cu T, sau fals (false), pe care o notăm cu F.

V = $\mathbb{L}$ (mulţimea valorilor logice) = {T, F};

O = $\mathcal{L}$ (mulţimea operatorilor logici)

Tipul şir de caractere

- Tipul şir de caractere a fost implementat pentru reprezentarea unei mulţimi ordonate de caractere care este tratată ca un tot unitar.

V = $\{\mathbb{P}_C\}$ (mulţimea caracterelor care este formată din litere, cifre și caractere speciale);

O = $\mathbb{C} \cup \mathbb{R}$ (mulțimea caracterelor și mulțimea operatorilor de concatenare)

În memoria internă, fiecare caracter din şir se reprezintă prin codul său ASCII:

- Constantele de tip şir de caractere se specifică prin mulţimea ordonată de caractere care compun şirul, delimitată în funcţie de limbajul de programare, de anumite semne speciale: apostrofurile ('Bună ziua') sau ghilimelele ("Bună ziua").
  - alfa - identificator de dată elementară
  - "alfa' – construcţie gresită
  - 'alfa" – construcție greşită
  - "alfa" - constantă de tip şir de caractere
  - 'alfa' - constantă de tip şir de caractere
  - "500" - constantă de tip şir de caractere
  - '500' - constantă de tip şir de caractere
  - 500 – constantă de tip numeric

- constanta de tip numeric 500 este diferită de constanta de tip şir de caractere "500", din punct de vedere al modului de reprezentare în memoria internă a calculatorului, din punct de vedere al operatorilor acceptaţi. De exemplu, asupra constantei numerice se pot aplica operatori matematici şi relaţionali, iar asupra constantei de tip șir de caractere, operatori de concatenare şi relaţionali. Constanta de tip numeric este reprezentată în memoria internă prin conversia în binar a numărului, iar tip şir de caractere este reprezentată prin conversia fiecărui caracter din șir în 8 cifre binare corespunzătoare codului ASCII al caracterului respectiv.

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL