ТЕМА № 1 ЕЗИЦИ И ИНФОРМАЦИЯ

1. **Информация**.

Произход на думата информация

Информацията (от латински: informare, „обучавам“) представлява налично, използваемо знание, но понятието се използва в сравнително широк кръг от значения.[1] Информацията може да се създава, унищожава, предава, приема, съхранява и обработва.

Понятието информатика означава според тълковния Речник на българския език (Издателство на БАН, София, 1990г.) наука за автоматично обработване на информацията. Речникът посочва, че думата е заимствана от френския език, като със същото звучене тя се среща още в немския и руския език. В английския език се използва терминът computer science (компютърна наука), но под натиска на европейските езици английски говорещите са склонни да допуснат съответното informatics. Във френския език думата INFORMATIQUE е получена от съчетаването на думите INFORMAtion (юп (информация) automaTIQUE (автоматичен, автоматика). За да разберем с какво се занимава науката информатика трябва да

изясним поотделно понятията информация и автоматична обработка.

2. Предмет на науката информатика.

Информатиката е една от най-младите и най-бурно развиващите се научни дисциплини. За начало на нейното оформяне като самостоятелна дисциплина могат да се приемат 30-те и 40-те години на XX век, макар че отделни нейни понятия, идеи и методи са се зародили много по-рано. По значимостта си за обществото и неговото усъвършенстване обаче тя стои редом с математиката и философията, тъй като трудно може да се намери област на човешката дейност, която да не е повлияна пряко или косвено от постиженията на информатиката. Нещо повече, могат да се посочат редица области на знанието и практиката, в които напоследък беше постигнат значителен напредък благодарение най-вече на достиженията на информатиката.

Познаването на основите на информатиката вече не е въпрос на личен избор, а насъщна необходимост за всеки човек. Не случайно това познаване се означава с понятието компютърна грамотност. Само преди 20 години, за да се нарече един човек грамотен, беше достатъчно да може да чете и пише на родния си език и да смята с естествени числа. Днес понятието грамотен предполага и възможността да си служиш с компютър в ежедневните си занимания.

Науката информатика разглежда информацията независимо от конкретния и смисъл.

Информацията е основно понятие не само в информатиката а изобщо в живота на хората. Това е така защото хората общуват, обменят информация, за да могат да се разбират по между си.

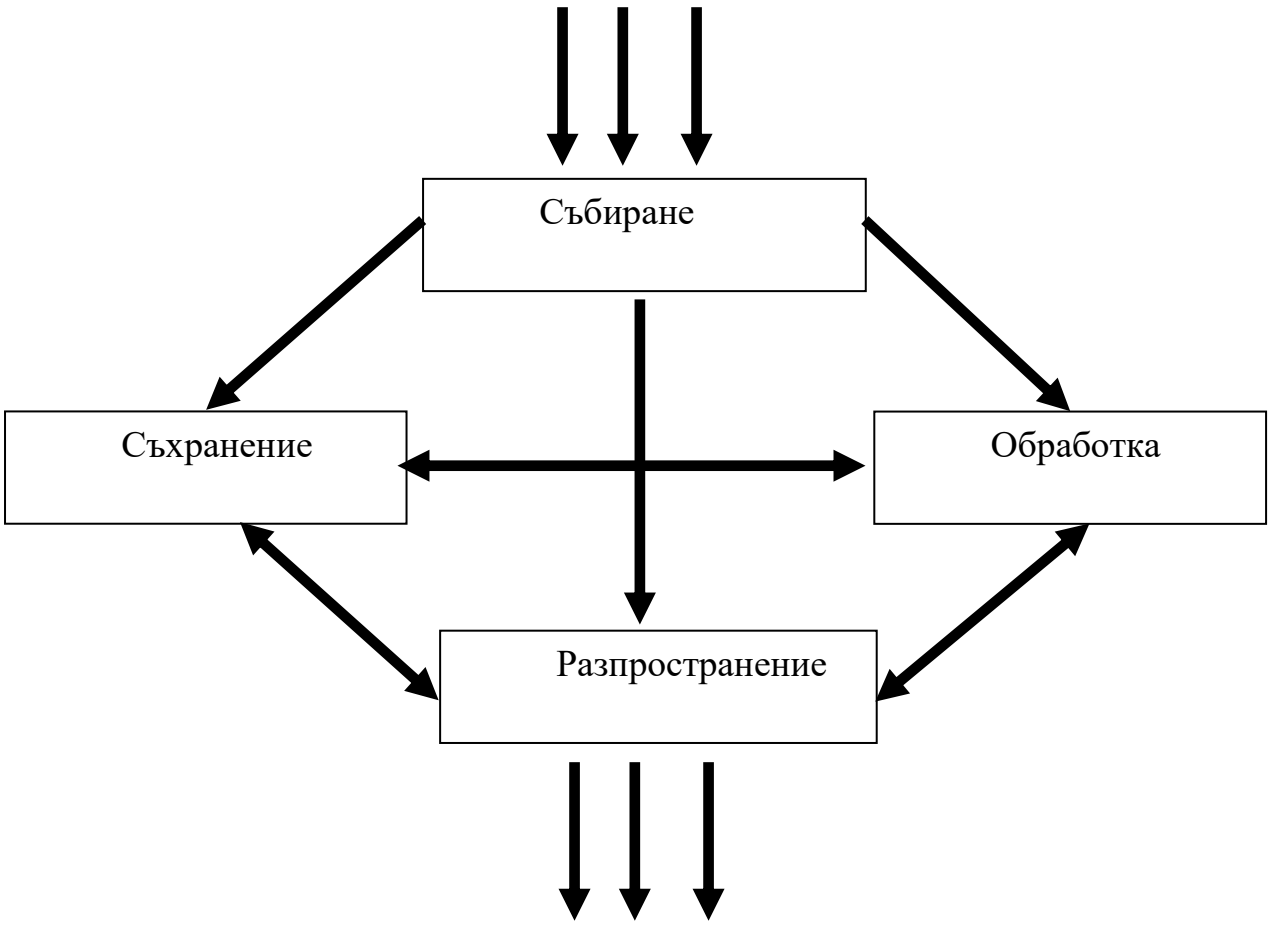
За да се разбират добре те трябва да използват общ език за комуникация по между си. С други думи трябва да е налице система от знаци и правила с помощта на които да изразяват своите мисли. За комуникация между хората са известни много езици, те ас възникнали преди много години, използват се и сега. Това с т. нар естествени езици. Те се характеризират с това, че съществуват в писмена и говорима форма. Което означава, че хората обменят информацията чрез текст, звук, графика и т . н.

Не на последно място е една система от знаци – числата, т.е информация която обменяме с помощта на числата. Този език е може би един от най-древните. С него се развива културата за смятане и стремеж за автоматизирано пресмятане.

В годините на еволюцията на човешката цивилизация се развиват не само естествените езици, но и т. нар. изкуствени езици.

3. Основни информационни дейности

- Събиране
- Съхраняване
- Обработка
- Разпространение



4. Основни характеристики на информацията.

- Достоверност
- Значимост
- Пълнота
- Съдържателност
- Актуалност
- Достъпност

5. Примери за изкуствени езици

- Езикът на математиката
- Езикът на химията
- Езикът на физиката
- Езикът на числата
- Езикът на пътните значи
- Езикът на музиката (нотните знаци)
- Езикът на геогтрафските карти
- Езикт на Морз
- Езикът на баркодовете

Въпроси и задачи към темата:

1.

ТЕМА № 2 ЕЗИЦИ И ГРАМАТИКИ

1. **Азбуки**.

Както естествените, така и изкуствените езици имат своя азбука и граматика.

Азбуката е крайно непразно множество от елемнти, наричани знаци, букви на азбуката.

Например: {1, 2, 3, … 9, 0}

{a, b, c, d, e, f, …}

2. Дума.

Елементите на всяка азбука се наричат букви или знаци. С помоща на буквите можем да образуваме думи.

Нека А е произволна азбука. Всяка крайна последователност от букви /знаци/ на дадената азбука се нарича дума. При така дефинираното поняти дума /низ/ се знае или е известно коя буква е първа, коя втора и т. н.

Думата се характеризира с дължина. Дължината представлява броя на знаците /буквите/. Дума, която не съдържа нито една буква /знак/ се нарича празна дума или празен низ.

Под дума на дадена дума се нарича последователност от букви /знаци/ от дадената дума.

Например: A=a_1 a_2 a_3 a_4 a_5 a_6 е дума.

Една неина поддумае: a_3 a_4 a_5

Напишете други поддуми на дадената дума.

3. Операции с думи.

- Конкатенация
- Копиране на думи и подуми
- Търсене на поддуми
- Вмъкване на поддуми
- Изтриване на поддуми
- Сравняване на поддуми и думи.

Въпроси и задачи към темата.

ТЕМА № 3 ГРАМАТИКА НА ИЗКУСТВЕНИТЕ ЕЗИЦИ

1. Граматика на естествените езици.

Граматиката на естествените езици представлява множество от правила, по които се изграждат различни езикови конструкции - думи, изречения.

2. Синтаксис и семантика.

а) синтаксис – правила за писане, правилно писане.

б) семантика – смисъл на това което е правилно написано.

3. Граматика на изкуствените езици.

Граматиката на изкуствените езици е заимствана от граматиката на естествените езици, но те са ограничени и се свеждат само до синтаксис и семантика на изкуственитъе езици.

4. **Метаезици**.

За да може един изкуствен език да бъде подложен накомпютърна обработка, трябва неговия синтаксис да бъде много точно и ясно определен.

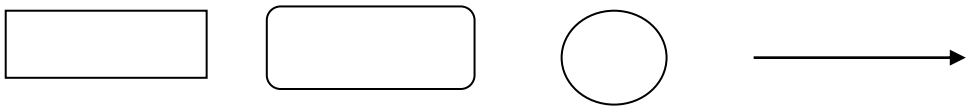
Това се налага поради необходимостта от това компютъра да може да разпознава синтактично правилните от синтактично неправилните езикови конструкции т.е думи и изречения. С други думи, кои от тях са елементи на дадения език и кои не са.

За описание наизкуствените езици се използват други езици – наречени метаезици.

Всеки метаезик има своя азбука. Обикновенно азбуката на метаезика се състои от някоко знака.

С помощта на метаезиците можем да описваме синтактични правила за конструиране на правилни изречения.

Графични символи на метаезика на синтактичните диаграми:



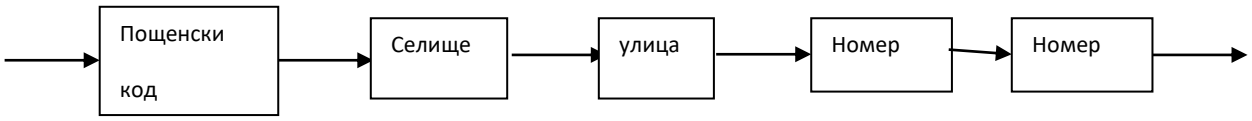
5. Видове конструкции.

При мета езиците има три вида конструкции:

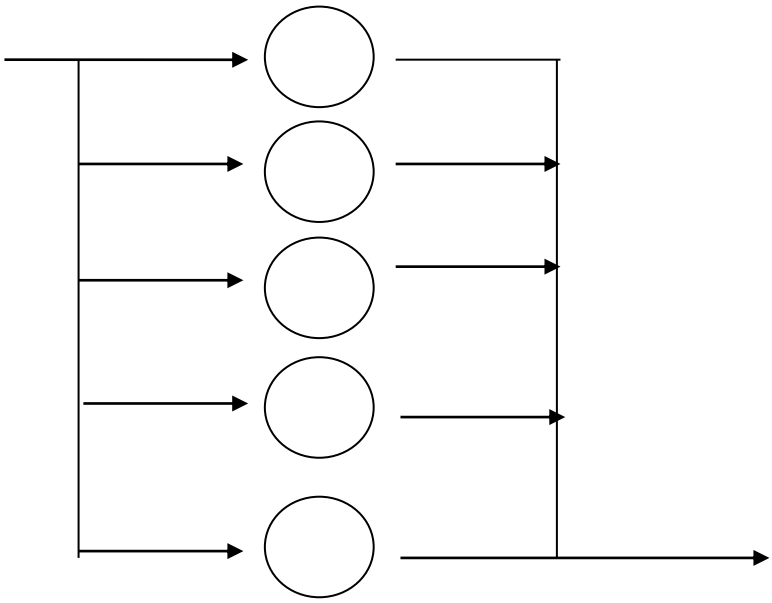
- Конструкция последователнос
- Конструкция алтернатива
- Конструкция повторение

Примери:

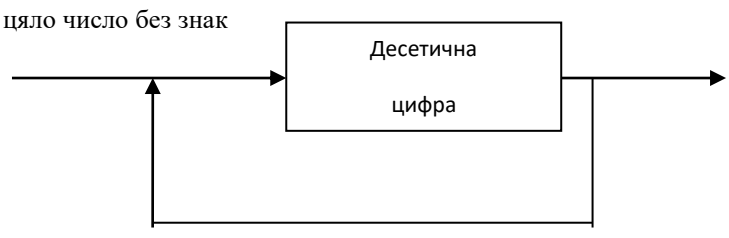
- Дефиниция на понятието адрес.



- Дефиниране на понятието оценка.



- Дефиниране на понятието цяло число без знак.



Въпроси и задачи към темата:

ТЕМА № 4 БРОЙНИ СИСТЕМИ

1. Общи сведения за бройни системи.

По бройна система се разбира метод за представяне на произволно число посредством съвокупност от цифрови знаци и правила за записването им.

Още от дълбока древност хората са имали начини да броят, смят и записват. Също така те са извършвали и основните аритметични дейстия а именно; събиране, изваждане, умножение и деление.

2. Непозиционни бройни системи.

Непозиционните броини системи се характеризират с неограниченият брои цифри /знаци/, които се използват при представянето на числата, както и факта,че стойността на всяка цифра не зависи от позицията /мястото/ й в записа на числото.

Обикновенно тези системи се използват за номериране. Типичен пример за така система е римската бройна система.

Римски Цифри	I	V	X	L	C	D	M

Десетични цифри	1	5	10	50	100	500	1000

3. Правилза за записване на числата в римска броина система.

3.1 Стойността на числото, което се състои от еднакви цифри, се определя чрез събиране на стоинсотта на цифрите.

Пример: XXX = X+X+X=10+10+10=30

3.2 Стойността на число, което се състои от различни цифри, като по-младшите следват по старшите, се определя чрез събиране.

Пример: XXVI=X+X+V+I=36

3.3 Стойността на число, което се състои от различни цифри, акто по-младшите прдхождат по старшите,се определя чрез изваждане на стойността на по-малдшата от стойността на по-старшата.

***Пример*: XIV = X+V-I=14**

4. **Позиционни броини системи.**

При позиционните бройни системи имаме ограничен брой цифри за представянето на числата, като стоиността на

всяка цифра зависи от позицията /мястото/ йв записа на съответното число.

Броят на цифрите използван в позиционните бройни системи за представяне на числата се нарича основа.

Наименование	**Основа**	**Цифри за представяне**
Десетична	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Двойчна	2	0,1
Осмична	8	0, 1, 2, 3, 4, 5, 6, 7
Шастнадесетична	16	0, 1, ..., 9, A, B, C, D, E , F

Примери:

$1574_{(10)}$

$246_{(8)}$

$4АБ9_{(16)}$

$547_{(10)} = 5.10^2+4.10^1+7.10^0$

Теорема: Некаизберем едно положително число $p \geq 2$ за основа на позиционната система.

Всяко естествено число А може да се представи, и то по единствен наачин във вида:

$$A = a_n p^n + a_{n-1} p^{n-1} + ... + a_1 p^1 + a_0 p^0$$

$$A = \sum_{k=0}^{n} a_k p^k$$

Нъдето коефициентите $a_к$ (к=1, 2, 3, ... n) са цели числа и удовлетворяват условията $0 \le a_k \le p-1$, k=0, 1, 2, … n. Тогава последователността $a_n a_{n-1} a_{n-2} \ldots a_1 a_0(p)$ яе нарича запис на естественото число А в р-ична позиционна систе.

Въпросии задачи към темата:

ТЕМА № 5 ПРЕВОД НА ЧИСЛАТА ОТ ЕДНА В ДРУГА БРОИНА СИСТЕМА

1. Превод на числата от десетична бройна система в двойчна бройна система.

Азбуката на числата в двойчна бройна система е {0, 1} т.е с основа 2. Нека числото $A_{(10)}$ е цяло положиотелно число. Да преведем десетичното число $A_{(10)}$ в двоична бройна система, означава да намерим такова двойчно число $B_{(2)}$, е двете числа да са равни $A_{(10)}=B_{(2)}$.

Пример:

$A_{(10)} = 302$

302 : 2 = 151 остатък 0

151 : 2 = 75 остатък 1

75 : 2 = 37 остатък 1

37 : 2 = 18 остатък 1

18 : 2 = 9 остатък 0

9 : 2 = 4 остатък 1

4 : 2 = 2 остатък 0

2 : 2 = 1 остатък 0

1 : 2 = 0 остатък 1

Записваме остатъците в обратен ред /от последния към първия/.

$B_{(2)}=10010111_{(2)}$

От разгледаия пример можем да изведем формула за общия случай:

Нека $A_{(10)}=a_к10^к+a_{к-1}10^{к-1}+...+a_110^1+a_010^0$, трябва да приведем /представим/ във вида:

$A_{(10)}=b_m2^m+b_{m-1}2^{m-1}+...+b_12^1+b_02^0$

Таблица на двойчните числа до 15;

0	0000	6	0110	12	1100
1	0001	7	0111	13	1101
2	0010	8	1000	14	1110
3	0011	9	1001	15	1111
4	0100	10	1010		
5	0101	11	1011		

Примери:

а) $1111_{(2)}=1.2^3+1.2^2+1.2^1+1.2^0=8+4+2+1=15$

б) $1001011110_{(2)}=302$

в) $1000000011_{(2)}=515$

г) $10000000000_{(2)}$

За по-лесно пресмята меже да се използва следната таблица:

2^0	1	2^5	32	210	1024
2^1	2	2^6	64	2^{11}	2048
2^2	4	2^7	128	2^{12}	4096
2^3	8	2^8	256	2^{13}	8192
2^4	16	2^9	512	2^{14}	16384
				2^{15}	32768

Въпроси и задачи към темата:

ТЕМА № 6 СЪБИРАНЕ, ИЗВАЖДАНЕ, УМНОЖЕНИЕ И ДЕЛЕНИЕ НА ЧИСЛА В ДВОЙЧНА БРОИНА СИСТЕМА

1. Събиране на числа в двойна бройна система.

Правила за събиране на числа в двойна бройна система.

+	0	1
0	0	1
1	1	10

При събиране на числата в двойна бройна система се започва с цифрите от най-дясната позиция.

Примери:

```
                100101
a)          +
                 10110
      ------------------
                111011
```

б)
```
  1011101
+
   110101
---------
 10010010
```

в)
```
    111
+
   1011
---------
   1111
```

2. Изваждане на числа в двойчна бройна система.

0 + 0 = 0	0 – 0 = 0
0 + 1 = 1	1 – 1 = 0
1 + 0 = 1	1 – 0 = 1
1 + 1 = 1 0	10 – 1 = 1

3. Умнможение на числа в двойчна бройна система.

0 * 0 = 0
0 * 1 = 0
1 * 0 = 0
1 * 1 = 1

Правилото за умножение да числа в двойчна бройна система е аналогично на това, как се умножават числа в десетична бройна система.

Пример:

```
1101*101
--------
   1101
  0000
 1101
-------
1000001
```

4. Деление на числа в двойчна бройна система.

Деклението на числа в двойчана брпойна система е аналогично на делението на числа в десетична бройна система.

Въпроси и задачи към темата:

ТЕМА № 7 КОДИРАНЕ НА ИНФОРМАЦИЯТА

1. **Данни**.

Данните или информацията с която общуваме може да се представи под различни форми: числа, букви, чертежи, графики, картини, точки, знаци. и т. н

Особен интерес представлява информацията, която се изразява с краен брой знакове от предварително определено множество.

Така представената информация се нарича дискретна. Процеса на представяне или преобразуване се нарича дискретизация.

2. Свойства на дискретната информация.

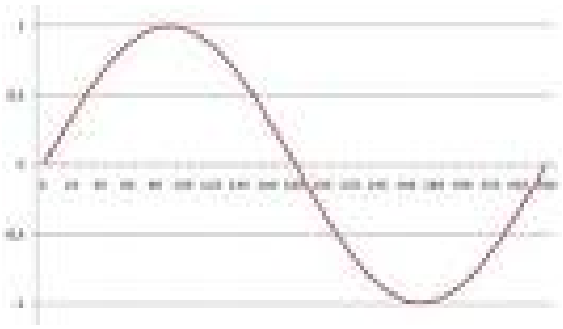
а) тя е устойчива къмвъншни влияния, може да се предава на разстояния и да се съхранява без изменение.

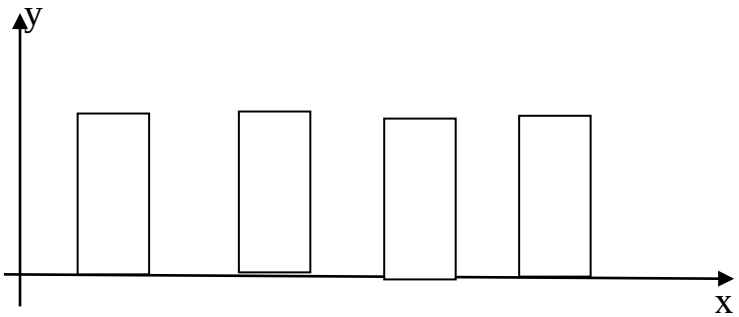
б) удобна е за автоматична обработка.

Поради тези две свойства /качества/ дискретната информация е основа за информатиката и компютърната техника.

Днес почти всичко е цифрова информация: сателитни телевизии, GSM, интернет.

Примери:





- Непрекъсната информация е музиката записана върху магнитен носител, кардиограмата на сърцето и др.

- Дискретна е информацията, представена чрез числа, ноти, чрез елементи нариани пиксели в растерните кадри на телевизорите или компютърните монитори. Като синоним на понятието дискретна информация широко се използва и понятието цифрова информация.

3. Измерване на информацията.

Освен мерните единици, които са ни известни за разстяни, тегло, височина, напрежение, ток и т н информацията също се измерва.

Основната единиза за измерванена информацията е1 бит /bit/.

1- бит информация представлява илисе нарича количеството информация определящо кое от две равновестни събития се е случило. 0 или 1

Например дали при хвърлянена един зар се паднало четен или нечетен брой точки, тура или ези при хвърляне на монета и т н.

Названието бит /bit/ произлиза отанглийското словосъчетание binary digit /двойчна цифра/.

По-голяма единица за измерване на информация е байт /byte/. 1 byte = 8 bits

Баитът е информация за изход на едно от 2^8 = 256 равновестни събития. Единицата бит и байт са основни единици за измерване на количеството информация в международната система СИ.

Освен тези основни единици за измерване на оличеството информация има и други, които се наричан пройзводни на основните единици.

Единица	K	M	G	T
Брой битове байтове	1024 2^{10}	1048576 2^{20}	1073741824 2^{30}	1099511627776 2^{40}

Има и други по-големи. Потърсете в Интернет.

4. Кодиране на информацията.

4.1 Код – наричаме множество от думи /наричани още кодови комбинации/, образувани от знаците на дадена азбука.

4.2 Кодиране – процес на представяне на информацията чрез определен код.

4.3 Декодиране - това е обратен процес на кодирането.

Текст .....КОДИРАНЕ.....текст 1
Текст1.....ДЕКОДИРАНЕ..... Текст

Предсатвянето на числата в различни броини системи са примери за кодиране в различни азнуки.

Начините за представяне на числата от една бройна система в друга бройна система задават начина закодиране и декодиране на числата съответно от една в друга БС.

4.4 Код на Юлий Цезар

Кодът на Юлий Цезар се състой в това,че всеки занак в оригиналния текст/съобщение/ се замества от втория следващ го знак на съответната азбука накоято е написано съответното съобщение.

При това предпослевдният знак се замества с с първия, а последният с втория знак на азбуката.

Например при кирилицата ще имаме:

А в оригиналното съобщение се замествва с В, Б – Г, .. , Ю-А, Я-Б.

При този начин на кодиране е очевидно, че декодирането възстановява еднозначно оригиналното съобщение.

Съобщението ‘QUO VADIS” записано с букви на латинската азбука се кодира в “SWQ YCFKU”. /На български къде отиваш?/

5. Приложение на кодирането.

В днешно време информацията се кодира по няколко причини.

- За да се записва кратко, с цел да не заема много място при съхраняване и за по бързо и по-евтино предаване на разстояние.
- За да се улесни извършването на различни операции при обработката й.
- За да се засекрети достъпа до информацията, в този случай се казва, че информацията и шифрована или криптирана.
- За да се защити от промяна при предаване и съхраняване.

Независимо каква кодова азбука използва всеки нейн знак може да се номерира с число, записано в двоична бройна система. Следователно всяка редица от кодови знаци /думи/ може да се представи с нули и единици.

6. Защита на информацията.

Защитата на информацията /данните/ при представяне под формата на съобщение сеосигурява като съдържанието на съобщението се подчини на точно определени правила.

При нарушение на тези правила се говори за изкривяване на информацията. /Например ако не се спази правилото за кодиране при кода на Юлий Цезар/.

Съответно имаме кодове за откриване на откриване на грешки и кодове за коригиране на грешки.

Въпроси и задачи към темата:

ТЕМА № 8 ДВУЗНАЧНА ЛОГИКА

1. Съждения.

Нека да разгледаме следните изречения.

- Две +две = 4
- Днес е слънчево
- Три е по-голям от 2

Тези изречения са такива, че за всяко може да се постави въпрос вярно ли е или невярно изказаното твърдение и да се очакват два възможно отговора – да или не. /истина или лъжа, вярно или невярно/

Изречение на естествен език, съдържанието на което може да се оцемнява като вярно или невярно /истина или лъжа/, се нарича съждение.

2. Прости и сложни съждения.

- Едно създения е просто ако не може да се разглежда, като съставено от други съждения.

Днес е слънчево

- Сложно съждения е такова съждения,което може да се разглежда, като съставено от други съждения. С други думи отделните компоненти на съждението са пак съждения. Т.е може дадено съждение може да се раздели на отделни компоненти – съждения.

Две е по-голямо от три или две е по-малко от 10

- ***Две е по-голямо от три***
- ***две е по-малко от 10***
-

3. Вярностна стойност на съждения.

а) Вярностна стойност на просто съждения.
Вярностната стойност на просто съждения предстявлява истринноста на съждението с други думу вярно или невярно /истина или лъжа/.

б) вярностна стойност на сложно съждения.
Вярностната стойност на сложно съжденията зависи от истинността на неговите компоненти /отделни съждения/.

4. Логика. Двузначна логика.

Терминът логика има гръцки произход /логос – дума, понятия, разум/ и се използва за означаване на общите закономерности на света и мисленето.

Математическа логика е наука за правилни математически разсъждения и изводи /правила за изводи/.

Съвременната математическа логика започва своето развитие от трудовете на Ирландския математик Джордж Бул /1815-1864/.

Логика в която дадено съждение е вярно или невярно се нарича двузначна логика.

5. Логически променливи и функции.

Съжденията подобно на променливите в алгебрата, ще означаваме с латинските букви.

Обикновенно стойността на едно съждения е: 1 когато съждението е вярно и 0,когато съждението е невярно.

Логически отношения.

- „И” $\wedge$
- „ИЛИ” $\vee$
- „НЕ” $\neg$

$\wedge$ - този знак се използва за означаване на операцията И, нарича се логическо умножение.

$\vee$ - този знак се използва за означаване на операцията И, нарича се логическо умножение.

$\neg$ - означава се логическо отрицание

Когато разглеждаме логически отношения подобно на алгебрични операции, можем да образуваме логически изрази.

(a $\wedge$ b) $\vee$ **(c** $\wedge$ d)

Стойностите на логическите изразизависят, т е са функции от участващите в тях логически променливи.

Както стойностите на променливите, така и стоиностите на функциите са само две – истина или лъжа /вярно или невярно, 0 или 1/.

Затова се казва, че логическите функции са двоични функции на двоични променливи /аргументи/.

x	0	0	1	1
y	0	1	0	1
xvy	0	0	0	1
x $\vee$ y	0	1	1	1
ако x то y	1	1	0	1

Таблица на логическите функции.

6. **Свойства на логическите функции.**

x $\wedge$ 1 = x	x $\vee$ 1 = 1
x $\wedge$ 0 = 0	x $\vee$ 0 = 0
x $\wedge$ $\bar{x}$ = 0	x $\vee$ $\bar{x}$ = 1
x $\wedge$ x = x	x $\vee$ x = x

7. Закони на Де Морган.

$$\overline{x \wedge y} = \bar{x} \vee \bar{y}$$

$$\overline{x \vee y} = \bar{x} \wedge \bar{y}$$

Теорема:

Всяка двойчна функция, независимоот това колко е броя на аргументите, може да се представи като логически израз, в които участват само трите логически операции; И, ИЛИ, НЕ.

Въпроси и задачи към темата:

ТЕМА № 9 АЛГОРИТМИ

Как да обясним някое сложно дейсствие на изпълнител, който може да изпълнява или може да върши само прости дейстия илиоперации?

Трябва да представим сложното действие чрез последователсност от прости действия,които изюпълнителят може да извърши или изпълни и които ще доведет /или след изпълнението ще постигнем/ до желан резултат.

1. Алгоритъм.

Точното /и разбираемо за изпълнителя/ описание на действията, които трябва да се извършят или изпълнят, за да се постигне определен резултат, се наричаалгоритъм.

Действието, което изпълнителят може да извърши /изпълни/ без допълнитлни пояснения, се нарича елемнтарно действие.

Извършването /изпълнениеето на едно елементарно действие /операция/ се нарича стъпка.

Алгоритъм е система от указания /команди/, които задава елемтарни действия, и реда на изпълнението им, за да се получи опредлен резултат.

Терминът алгоритъм произлиза от името на Абу Джафар Мохамед ибн Муса ал-Хорезми /арабски математика/, който около 820 г от н е написва трактат за това как да се представят /записват/ числата в 10-ична бройна система и как да се смята.

Отношение към даден алгоритъм има три категории:

- Съставител
- Изпълнител

- Потребител на алгоритъма.

  Потребителя задава изпълнението и ползва крайният резултат от изпълнението на алгоритъма.

Често като синоним на елемнтарно действие се използва заповед, команда или проста операция.

Всеки алгоритъм се характеризира с три посновни състояния:

- Начално състояние
  (входни данни, начални данни

- Обработк на данните
  (Алгоритъм)

- Резултат
  (Изходни данни, резултат)

Подалгоритъм е алгоритъм за изпълнение на типична задача /последователност от действия, команди, заповеди/, което се използва в даден алгоритъм.

2. Основни изисквания към алгоритмите.

- Да може да се представи сложното действие с помощ на по-прости действия, достъпни за изпълнителя,
- Да се използват само елемнтарни действия,
- Да бъдат описани ясно и точно последователността, в която трябва да се изпълняват елемнтарните действия.

Въпроси и задачи към темата:

ТЕМА № 10 СВОЙСТВА НА АЛГОРИТМИ

Алгоритмите имат различни характерни свойства. Нека разгледаме някои от тях:

1. **Дискретност** – описанието на всеки алгоритъм се състои от краен брой указания, а изпълнението на алгоритъма става в последователни, различни една от друга стъпки.
2. **Яснота** – разбираемост на алгоритъма означава, че изпълнителят може да извърши всяко текущо действие или стъпка от алгоритъма и да определеи еднозначно, коя е следващата стъпка, която трябва да изпълни.
3. **Формалност** – от изпълнителят не се изисква да знае каква цел се преследва с изпълнението на алгоритъма – той трябва да работо формално, да изпълнява указанията, докато достигне до укзание за край. Свойството формалност позволява изпълнението на алгоритъма да бъде и автомат – компютър.
4. **Определеност** – означава, че при всяко изпълнение на алгоритъма с еднакви входни данн ще се получава един и същи изходен резултат.
5. **Масовост** – дадем алгоритъм може да се прилага за решаване ан коя да е задача от даденклас еднотипни задачи. Сдруги думи един алгоритъм може да сеизпълнява и с различни входни данни /начални данни/ от някое допустимо множество от данни.
6. **Изпълнимост** – едно от важните изисквания, което се поставя на алгоритмите е той да се състои от изпълними стъпки.

7. **Крайност** – Изпълнителят на даден алгоритъм трабва да завърши след краен брой стъпки. В противен случай такива алгоритми не предсатвляват интерес за разглеждане. С други думи изпълнение на задачата в реално време. Резултатност.
8. **Ефективност** – алгоритъма да се изпълнява за приемлив брой стъпки и да не се налага да се позлва при работата си прекалено много място /памет/ за съхранение на даннитре. Една от важните задачи на компютърната информатика е да създава ефективни алгоритми.
9. **Измерване на ефективността** – с измерване на ефективността при изпълнение на даден алгоритъм се свързва броят на елемнтарните действия, също така и броят на входните и междинни терезултати.
10. **Сложност** – Сложността ан алгоритмите се изчислява с помоща на математически формули.

Пример за ефективни алгоритми:

А. 2^{32} = 2.(2.(2.(...2(2.2)...)) – тук имаме да извършим 32 умноженижя, като лелемтарни действия или операции

Б. 2^{32} = ((((2²)²)²)²)² – тук имаме да извършим 5 елемнтарни действия или операции.

Тези два примера илюстрират ефективността на алгоритъм от гледна точка на броя на елемнтарните операции /дейстия/.

Въпроси и задачи към темата:

ТЕМА № 11 ОПИСАНИЕ НА АЛГОРИТМИТЕ

1. Описание на алгоритмите.

Алгоритмите могат да се описват по няколко начина.

Един от начините за описание на алгоритми е словестният начин. Той не е особено популярен. Основната причина за това е неднозначнстта в тълкуването намнпого думи от естествените езици. За описанието на алгоритмите се използват обикновенно други начини.

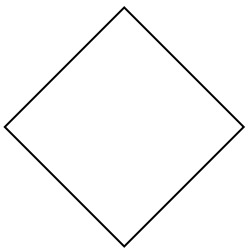
- Чрез блок-схеми
- Чрез езици за програмиране

2. Описание на алгортми чрез блок-схеми.

За да се създаде един алгоритъм с помоща на блок-схеми се използват следните основни блокове.

Начало, край

Вход, изходвъвеждат се данни
или се извежда рвзултат

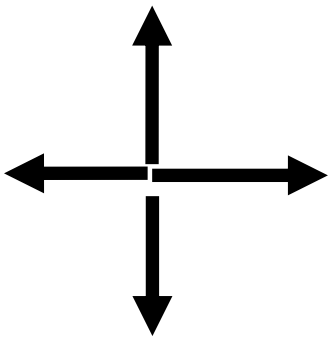


Решение, трябва да се избере една от две алтернативи. Нарича се условен блок

Обработка,посочва се една или група безусловни команди с които се извършва някаква обработка. Нарича се безусловен блок или команден блок

Подалгоритъм, извършва се обработка на подалгоритъм

Конектор, блок за връзка



Посока. Свързва два блока и показва последователността на тяхното изпълнение.

3. Описание на алгоритмите чрез език за програмиране.

За да може един алгоритъм да бъде изпълняван от компютър, той трябва да бъде написан със средствата на специални езици, известни като програми езици или езици за програмиране.

Текстът на алгоритъма,записан с помоща на език за програмиране, се нарича програма.

Изпълнението на всяка програма, написана на език за програмиране,най-напред се превежда на съответния език на компютъра – наречен машинен език.

Превеждането на програмите от програмен език на машинен език става с помоща на специални програми наречени транслатори.

Транслаторите не само извършват превод на машпинен език, те са в състояние да откриват и множество синтактиочни грешки в програмата.

Алгоритмите написани с помоща на евик за прорамиране допускат удобни означения за използване на велични, а програмите, написани на съответен език са лесно използваеми от различни ко,мпютри.

Днес едни от най-масовите езици за програмиране са: C, C^{++}, Фортан, Бейск, Паскал, Java, HTML, PHP, ...

Въпроси и задачи към темата:

ТЕМА № 12 ВИДОВЕ АЛГОРИТМИ

Както вече беше споменато в предните теми, алгоритъм е точно описание на действията, които трябва да се извършат от изшпълнителя а да се получи опеределен резултат.

1. Алгоритъм на Евклид.

Строго поределени правила за достигане на някаква крайна цел са били формулирани най-напред в математиката.

Още от дълбока древност е известен така нареченият алгоритъм на Евклд. Този алгоритъм служи за намиране на най-големият общ делител на две естествени числа.

Алгоритъм на Евклид за намиране на НОД

Вход: числа a и b, Изход; НОД (a, b)

1. Въведете и запомнете числата a и b.
2. $a \neq b$ изпълнете стъпка 3, в противен случай – стъпка 5.
3. Ако a > b, то намалете стоиността на a с b, в противен случай намете с стойността на b.
4. Изпълнете стъпка 2.
5. Съобщете стойността на a като резултат.
6. Прекратете работа.

Например при начални a=2 и b=6 последователно се получават двоиките (a,b):

(2; 6), (2, 4), (2,2). Така следкараен брой стъпки, получаваме, че НОД на 2 и 6 е 2.

В алгоритъма на Евклид се предполага, че входните числа са естествени /цели и положителни/ а в горното описание този момент не напира отражение.

Следователно ако този алгоритъм се изпълнява от компютърм като входни данни се подадат числата 3 и -3, то компютърът не би имал основани е да ги отхвърли. В този случай при так авъведените данни 3 и -3 алгоритъмъ никогфа няма да завърши.

За да бъде краен горният алгоритъм, трябва да се дошълни с предварителна проверка за входните данни, която проверка да не допуска те да не са естествени числа.

2. Параметри на алгоритмите.

Всеки алгоритъм задава начин за решаване на съответна задача.

Алгоритмите се различават по възможното множество от допустими входни данни, получаваните резултати и правилата за достигане на резултатите.

2.1 множество от възможни начални данни /входни/.

2.2 Множество от получавани крайни резултати /изходни данни/

2.3 Множество от междинни резултати

2.4 Правило за начало – как да започне изпълнението на алгоритъма/

2.5 Правила за обработка /как се получават междинните резултати/

2.6 Правило за край – как приключва изпълнението на алгоритъма/

2.7 Правило за позочване на крайния резултат.

3. Видове алгоритми.

Както вече знаем описанието на всеки алгоритъм се състой от отделни стъпки /команди/.

Командите задават изпълнението на елемнтарните действия или определят коя е следващата команда, която трябва а се изпълни.

В зависимост от типа на съответстващите ги команди алгоритмите се делят на три основни вида.

3.1 **Линейни алгоритми** – алгоритъма се състой от команди, които се изпълняват последователно една след друга по реда на записването им.

3.2 **Разклонени алгоритми –** тези алгоритми съдържат команди, които определят кои са следващите за изпълнение команди. Тези алгоритми позволяват изпълнението на алгоритъма да се управлява в зависимост от плучените до момента резултати.

3.3 **Циклични алгоритми –** тези алгоритми съдържат група от команди, които се изпълняват многократно. Тези действия се наричат повтарящи се или самото повторение на командите се нарича цикъл.

Обикновенно алгоритмите имат смесен характер.

Алгоритмите могат да се класифицират и по вида на входните данни:

- Числени
- Текстови
- Графични
-

Въпроси и задачи към темата:

ТЕМА № 13 ВЪВЕДЕНИЕ В ПРОГРАМИРАНЕТО

След като бъде създаен един алгоритъм за решаване на дадена задача /даден тип задачи/, неговото изпълнение може да се възложи на изпълнител.

Ако алгоритъмъ е написан добре и изпълнителят следва указанията, след краен брой стъпки ще получим резултат.

Компютърът е назаменим изпълнител на алгоритми. Това е така поради факта възможността му да изпълнява алгоритъма с голяма скорост и без гршки стъпките на алгоритъма.

Компютърната програма представлява алгоритъм във форма и вид, които могат да се възприемат и изпълняват от компютър.

За да може един алгоритъм да бъде възприет и изпълняван от компютър тойтрябва да е написан на език разбираем от компютъра с други думи да е на машинен език /машинна програма/.

1. **Хронология на езиците за програмиране.**

- **Първо поколение –** машинни езици, това са първите езици. Збуката им се състови от два знака 0 и 1.не са удобни за използване.

- **Второ поколение –** асемблерни езици, при тези езици се въвеждат символни означения. Адресите от паметта и

величните се цитират с имената.Тоз език е машинно зависим.

- **Трето поколение –** езици от високо ниво, тези езици се появяват втората половина на 50-те години. При тях е налице относително висока независимост от хардуера, при което се създава условие за преносимост. При тези езици команда /оператор/ се транслира /превежда/ в няколко машинни команди. Величниите и командите се записват на естествен език. Представители на тези езици са: Фортрам, Алгол. Кобнол, Паскал, С, С++ и др.

- **Езици от четвърто поколение –** тези езици са предназначени за изграждане на прложни системи. Тези евзици имат лесен интерфейс. Към тези езици спада СУБД, система за управление на ЕТ. и т н.

- **Езици от пето поколение –** тези езици се използват за разюработване на ситеми в областта на изкуствения интелект. Предствител – Пролог.

2. Език за програмиране.

Език за програмиране е език, предназначен за записване и разпространяване на компютърни алгоритми.

Езиците за програмиране имат няколко особени и важни изисквания:ю

- Трябва съответният Еп за е удобен за потребителите

- Програмите написани на ЕП трябва да са лесно и удобно възприемани и преобразуват от компютрите.

3. Транслатори.

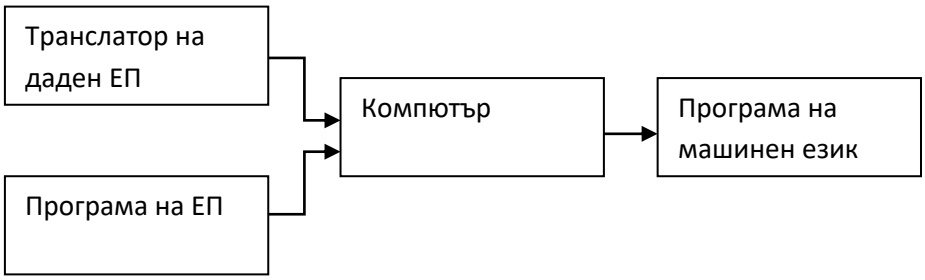
Трансалторът е компютътрна програма, предназначена да преобразува /превежда/ програми написани на определен език за програмиране в програма на машинен език.

Програмата получена в резултат на трнслацията се нарича изпълнима програма.

4. Етапи при изпълнение на програма.

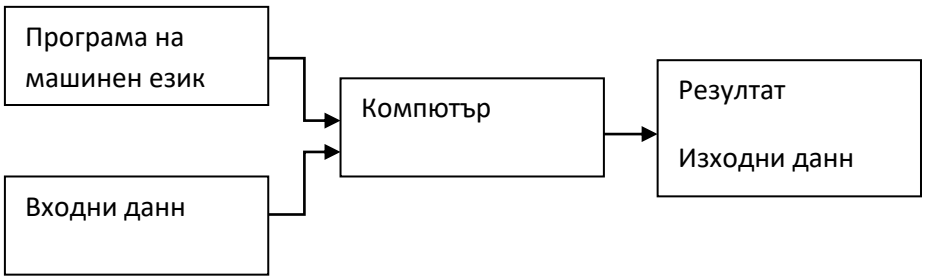
а) ЕТАП 1.

Транслиране на програма



б) ЕТАП 2

Изпълнение на програма



Обикновенно програмите се тестват с входни данни за които е известен резултата, които ще се получи.

Въпроси и задачи към темата:

ТЕМА № 13 СИНТАКСИС И СЕМАНТИКА НА ЕЗИЦИТЕ ЗА ПРОГРАМИРАНЕ

1. Азбука.

Азбука наричаме непразно множество от знаци.

Азбуката на повечето езици за програмиране включва: латинските букви, десетичните цифри и специални символи.

При съставянето на програма рябва да се спазват строги правила.

2. Синтактични правила.

Синтактичните правила дават правилата за писане на думи. Определят кои последователности от знаци на азбуката на съответния език за програмиране са допустими езикови конструкции т е имаме правилни и неправилни езикови конструкции.

3. Семантични парвила.

Определят смисъла на синтактично правилните езикови конструкции. С други думи определят как трабва те да се разбират от човека и как ще бъдат интерпертирани от компютъра при изпълнение.

4. Елементи на езиците за програмиране.

а) **служебни думи** – това е съвокупност от думи например за езика Паскал това са: until, while, do, if, for, .

б) **идентификатори** – последователност от букци и цифри задължително започваща с буква.

в) **изрази** – това е същототоняти, което е познато от математиката за израз.

г) структура на програмата – ВХОДНИ ДАННИ, ПРАВИЛА ЗА ОБРАБОТКА, РЕЗУЛТАТ/ИЗХОДНИ ДАННИ/

д) описание на данните – описват се типовете данни, които ще се използват – цели числа, реални числа, булев тип

е) описание на обработката – тук се описват праилата за обработка на данните или с други думи това са така наречените оператори на ЕП.

ж) Оператори – оператора задава определено елемтарно дейстие или операция.

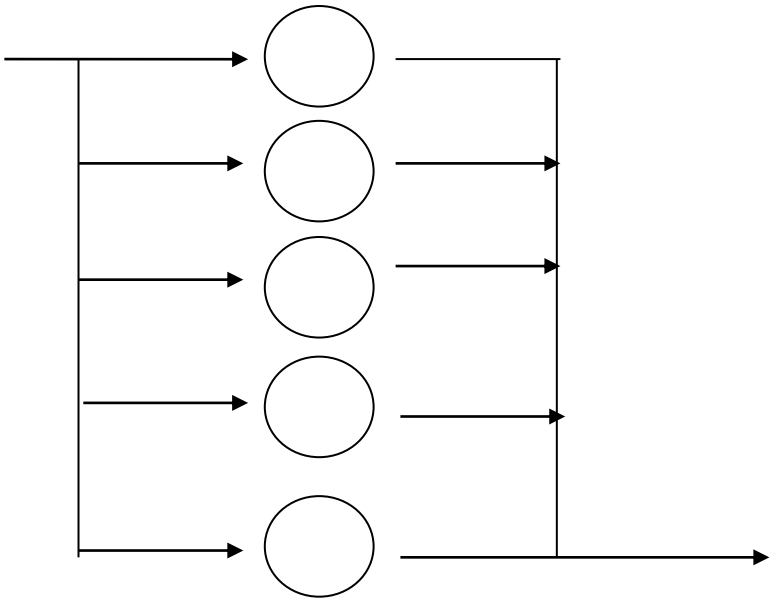
з) подпрограма – ако алгоритъма е сложен и е създаен така, че да се използва и подалгоритъм, то подалгоритъма се оформя като подпрограма. Структурата й наподобава тази на програмата.

и) Кометари – за по-лесна разбираемост за хората в текста на програмата могат да се вмъкват по определени правила така наречените коментари. Те съдържат обяснение за хората, но те не се вземат под внимание от транслатора.

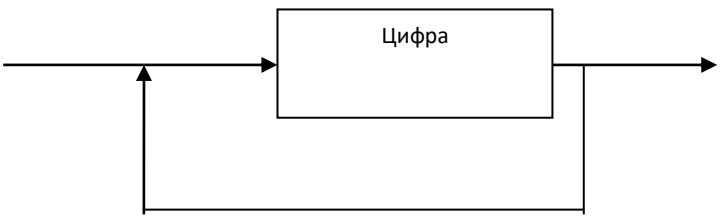
й) синтактична диаграма – графично средство за точно представяне на допустимите последователности от знаци в синтактичните конструкции на даден ЕП.

Примери:

- ***Десетична цифра***



- ***Цяло число без знак;***



Структура на програма на Паскал:

Заглавие	Program suma1
Раздел за деклариране на:	var
Константи	Constant
Променливи	Integer, real, boolena
Типове	type
Процедури	procedure
Функций	function
Начало	Begin
Изрази	
Команди	
Оператори	
Край	End.

Въпроси и задачи към темата:

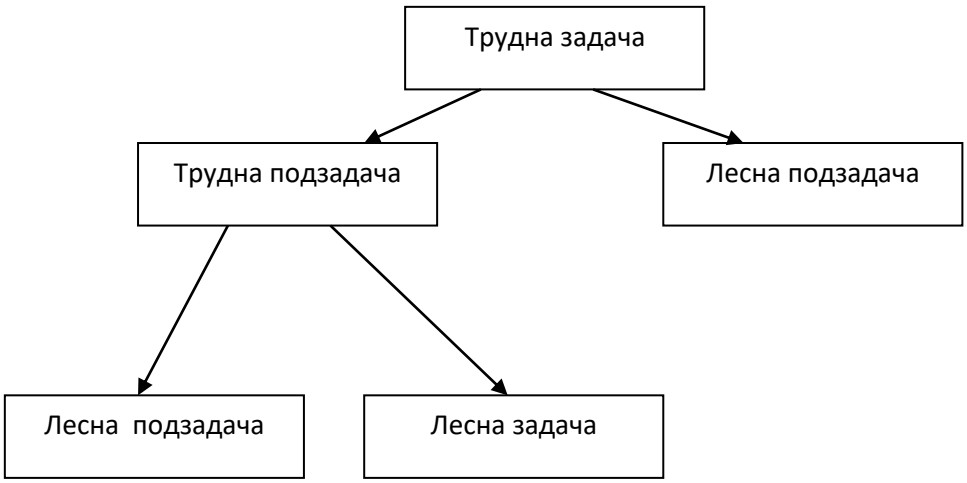
ТЕМА № 14 ТЕХНОЛОГИЯ НА ПРОГРАМИРАНЕТО

1. Модулно проектиране на програмите

Метод „Разделяй и владей”

При решаване на трудни задачи обикновенно я разбиваме на подзадачи, които в определен смисъл са по-лесни от първоначалната задача.

Ако е необходимо това можем да го приложим и за подзадачите, докато стигнем подзадачи, които можем лесно да решим.



Този метод е в основата си на тка нареченото модулно проектиране на програми, което се прилага при проектиране на структурата на сложни програми.

Съсъставянето на сложни компютърни програми се извършва чрез „разлагането”им на отделни модули – подпрограми.

По този начин се цели да се стигне до подпрограми, които са елемнтарни за създаване и обозрими за лесна проверка, или такива които са налични в библиотеките от подпрограми.

Модулното проектиране се прилага в два варианта – възходящо и низходящо проектиране.

1.1 Възходящо проектиране.

При възходящото проектиране /отдолу нагоре/ се започва с модулите от най-ниско ниво. Този процес продължава до достигане на модулите от най-високо ниво.

1.2 Низходящо проектиране.

При низходящото проектиране /отгоре надолу/ се запозва с модулите от най-високо ниво.

Това са всъщност основните модули на проекта и до голяма степен те отразяват основните функции на програмата. Тези модули се разлагат на други модули и т н.

Предимства на модулното проектиране:

- В разработката могат да участват повече от един специалист;
- Облекчава се четенето на програмите;
- Облекчава се прояверката на програмите;
- Облекчава се внасянето на промени в програмата;

В заключение може да се каже, че разработката се извършва по-лесно, по-бързо и е по-надеждна.

2. Етапи при разработка на програми.

При разработката на програмите имаме три основни етапа:

2.1 Формулиране на задачата.

Този етап включва няколко стъпки: анализ, формулиране на изискванията /проектно задание/ и спецификация /формално описание/ на задачата.

При анализа на задачата се изеснява нейната същност, област на приложение и се прави опит да се отнесе към съществуващи подобни решени задачи. Анализът завършва с формулиране на конкретна задача за програмиране.

Проектното задание включва изисквания към функционалните и експлоатационните характеристики на програмата.

Формалното описание на задачата най-често представлява описанието й със средствата на формален език, например със средствата на езика на математиката.

2.2 Проектиране и програмиране .

При този етап се определя модулната структура на програмата и се продължава с програмната реализация и тестване на отделните модули на програмата.

- Определяне на модулната структура на програмата;
- Избор на алгоритъм и оценка на всеки отделен модул;

- Кодиране /записване на алгоритмите на съответния език за програмиране/ и документиране;

- Компилиране и поправка на синтактичните грешки – ако има;

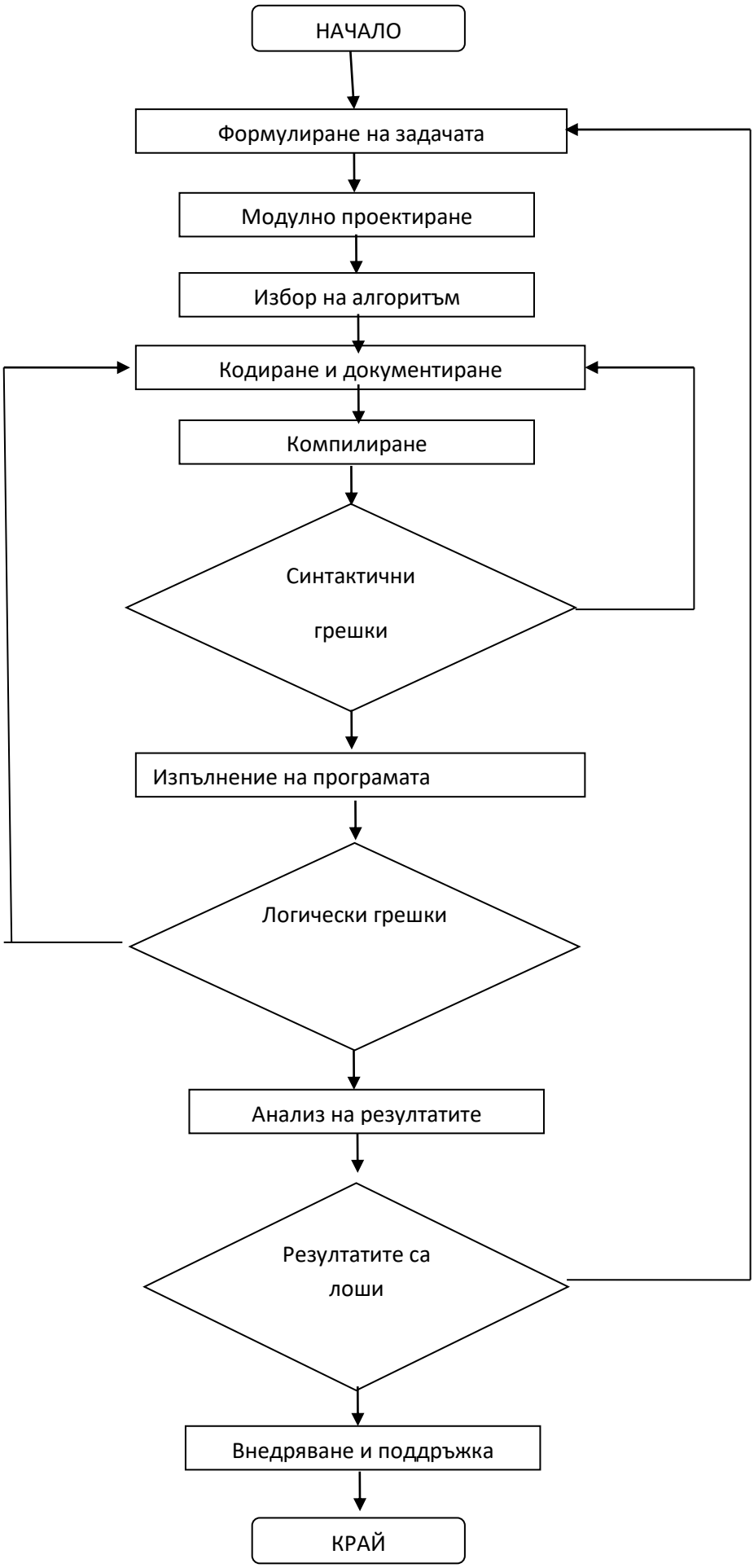
- Тестване и поправка на логиески грешки;

2.3 Избор на алгоритъм.

Изборът на алгоритъм е много важна стъпка в процеса на изграждане на цялата програма. Едни от най-ценните качества на един алгоритъм са времето, което е необходимо за изпълнението му, т.е скоростта наизпълнение, и паметта, която е необходима за представяне на използваните велични /структури от данни/.

Скороста на изпълнението на една програма се измерва с броя на операциите, които трябва да се извършат, за да се достигне до получаване на крайния резултат.

Оценката на алгоритъма се прави въз основа на това какъв е броя на операциите и съответното количество на входните данни.


НАЧАЛО
Формулиране на задачата
Модулно проектиране
Избор на алгоритъм
Кодиране и документиране
Компилиране
Синтактични грешки
Изпълнение на програмата
Логически грешки
Анализ на резултатите
Резултатите са лоши
Внедряване и поддръжка
КРАЙ

Пример:

Да се съствави алгоритъм за пресмятане на сумата на числата: 1+2+3+...+n.

Иешаването на тази задача може да се извърши по два начина:

I начин:

Входни данни: n – чило положително число,ю

Изходни данни: /резултат/ S – сумата на първите n естествени числа.

1. Въведи n
2. S=0
3. За I = 1 до n

   3.1 S=S+i
4. Изведи S
5. Край.

II Начин.

1. Въведи n
2. Изведи n*(n+1)/2
3. Край.

При първиет алгоритъм тялото на жикъла ще се изпълни n-пъти, а в него ще се извършат две операции: събиране и присвояване т е общият брой на операциите ще е 2n. С понеже извън цикъка има още три операции то общият брой на операциите ще бъде 2n=3.

При вторит алгоритъм броят на операциите е константа – 4.

С това се вижда, че втория алгоритъм е по-добър от първият.

Документиране на програмата.

Едновременно със създаването на програмата е необходимо да се създава и необходимата документация. Текста е свързан с описание на текстовете на прграма. Тоз текст или обяснение съдържа най-често, описание на програмата, предназначение на програмата, входни данни, какви изходни данни се получаван какъв е използвания алгоритъм

Тестване на програмата.

Тъй като е възможно при стартиране на прграмата да възникнат някои грешки, затова се налага тя да се тества или тестването още се прави с цел да се провери коректността на прграмата, как тя работи, това се прави обикновенно с известни входни данни и съответни за тези данни е ясно какъв резултат трябва да се получи.

Също така програмата трябва да се провери за системни и синтактини грешки.

Внедряване и поддръжка.

Внедряването или реалната реализация на прорамата е крайната цел. Възможно е по време на работа на програмата да възнукнат някои грешки, тесе отстраняват, затова се казва, че има поддръжка на прграмата.

Въпроси и задачи към темата:

ТЕМА № 15 ВЪВЕДЕНИЕ В ЕЗИКА ПАСКАЛ

1. Речник на езика Паскал.