

Увод в програмирането

Въведение в компютрите и
програмирането

Учебник

- М. Тодорова. Програмиране на С++ - първа част, С., СИЕЛА СОФТ ЕНД ПАБЛИШИНГ, 2002.
- **Moodle**
<http://moodle.le.tsdoit.org/course/view.php?id=19>
- Ключ за записване: **up3001**

Компютър

- **Компютърът е устройство или система, която е годна да изпълнява редица от операции по точно определен начин.**

Операциите са често числови изчисления, аналитични преобразувания или обработка на данни, но включват също и входно/изходни операции.

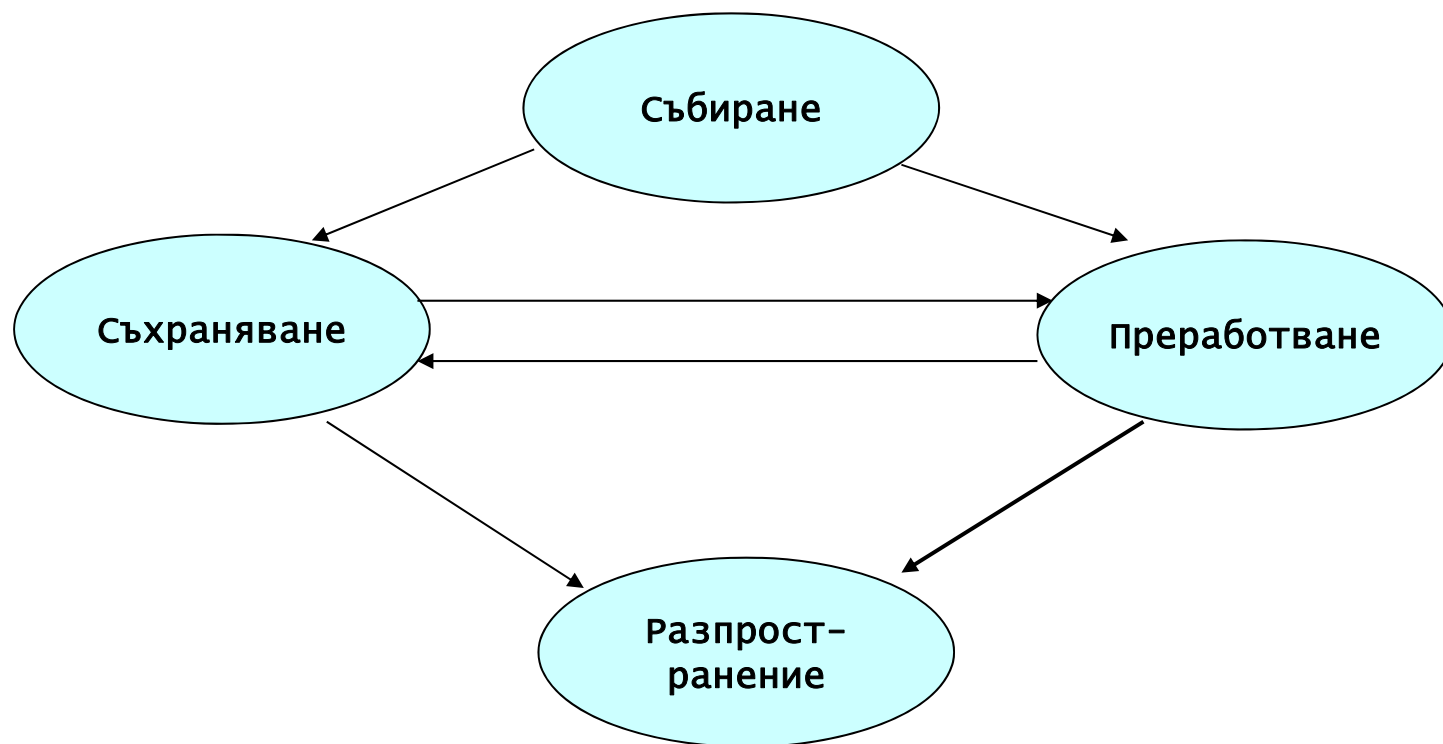
Чрез компютрите се извършват: текстообработка, графика, счетоводство, управление на устройства, телекомуникационни и банкови услуги и др. дейности.

- **Най-общо, компютърът е средство за представяне и обработване на информация.**

Информация

- Информацията е съвкупност от символи, над които се извършват следните информационни дейности: *събиране, съхраняване, преработване и разпространение.*

Информационни дейности

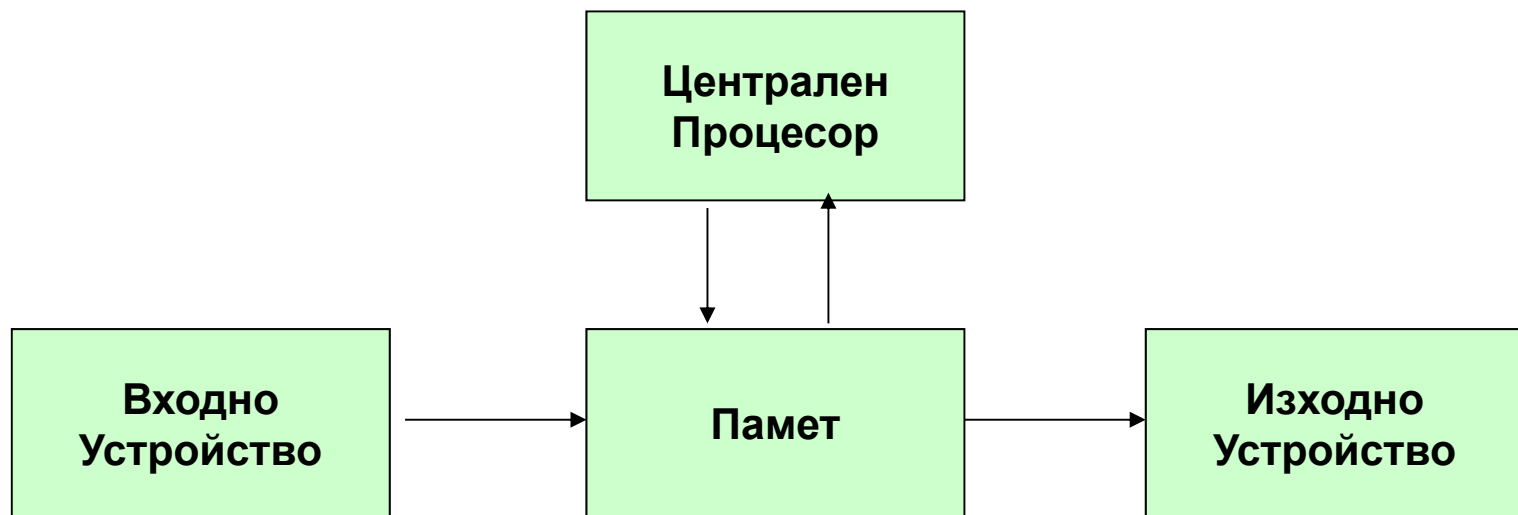


Поколения компютри

- *нулево поколение* – електронно механични устройства
- *първо поколение* – лампови ЕИМ - ENIAC (Electronic Numerical Integrator And Computer) е първия използваем компютър. Проектиран е от Дж. Преспър Екерт и Дж. Мокли от университета в Пенсилвания и е завършен през 1946 година.
- *второ поколение* – транзисторни ЕИМ (1948)
- *трето поколение* – компютри на базата на интегрални схеми (миникомпютри)
- *четвърто поколение* – компютри на базата на големи интегрални схеми (микрокомпютри)
- *пето поколение* - ?

Архитектура на изчислителна машина

- Джон фон Нойман (1946)



Памет

0	1	1	0	...
---	---	---	---	-----

- Паметта на компютъра може да бъде изобразена като редица от елементи, всеки от които е носител на информацията – цифрите 0 и 1. Такъв обем информация се нарича **бит**.
- Технически не е възможно да бъде реализиран пряк достъп до всеки бит от паметта, поради което няколко бита се групират в **дума**. Думата може да е с размер 8, 16, 32, 64 и т.н. бита.
- 8 – битовата информационна единица се нарича **байт**.

Памет ...

- Обемът на паметта се измерва в байтове (B), килобайтове (KB), мегабайтове (MB), гигабайтове (GB) и терабайтове (TB), където:
 - $1 \text{ KB} = 2^{10} \text{ B} \approx 10^3 \text{ B}$
 - $1 \text{ MB} = 2^{20} \text{ B} \approx 10^6 \text{ B}$
 - $1 \text{ GB} = 2^{30} \text{ B} \approx 10^9 \text{ B}$
 - $1 \text{ TB} = 2^{40} \text{ B} \approx 10^{12} \text{ B}$.

Памет ...

- Обикновено се реализира пряк достъп до всяка дума на паметта. Всяка дума се свързва с пореден номер, наречен неин **адрес**. Номерацията започва от 0.

0	...
1	...
	...
999	...

Памет ...

- Два вида памет: **оперативна (ОП)** и **външна (ВП)**
- Два типа **ОП**: **постоянна (ROM)** и памет с произволен достъп (**RAM**)
- **ВП** осигурява по-евтино съхранение на информация, която се запазва при изключване на захранването. На тази памет информацията се съхранява във вид на **файлове**

Памет ...

- **Файлът** е организирана съвкупност от взаимно свързани данни. Изграден е от отделни части информация, наречени **записи**. Всеки файл има **име**, с което той е определен еднозначно в множеството от файлове.
- Най-често за ВП се използва **твърд диск** (hard disk). Той се състои от въртящи се плочи с магнитно покритие и глави за четене и писане.
- Често се използва и **флопидиск** или **дискета**
- Компакт дисковете CD ROM
- DVD дискове
- USB disk drive



Централен процесор

- ЦП (CPU) управлява цялостната работа на компютъра
- ЦП се грижи за управлението на системата, за аритметиката и прехвърлянето на данните. Той открива и изпълнява инструкции, извършва аритметични и логически операции, взема данни от паметта и външните устройства и ги връща обратно там

Централен процесор ...

- ЦП се състои от управляващо устройство (УУ) и аритметично-логическо устройство (АЛУ)
- УУ координира работата на компютъра. То изпраща разрешаващи и забраняващи сигнали към отделните устройства и по-такъв начин синхронизира работата на системата
- АЛУ изпълнява различни аритметични и логически операции (събиране, изваждане, умножение, деление, логическо събиране, логическо умножение, отрицание, сравнение и др.)

Централен процесор ...

- Наборът от аритметични и логически операции определя т. нар. **език на компютъра** или **машинен език**. Всеки компютър може да изпълнява инструкциите на своя машинен език.
- Първият модел IBM PC използва за ЦП микропроцесора Intel/8088. Този процесор може да обработва 16 бита (2 байта) информация едновременно. Тактовата честота на компютъра беше 4.77 MHz

Входни и изходни устройства

- Наричат се още периферни устройства
- Входните устройства служат за въвеждане на информация в паметта на компютъра.
Широко разпространени са:
 - ☐ клавиатурата,
 - ☐ мишката,
 - ☐ дисковите и лентовите устройства,
 - ☐ камерите,
 - ☐ скенерите

Входни и изходни устройства ...

- Изходните устройства извеждат резултатите в подходяща за възприемане форма. Най-разпространени са:
 - ☐ мониторите,
 - ☐ дисковите и лентовите устройства,
 - ☐ принтерът,
 - ☐ тонколонките.
- ЦП, RAM паметта и електрониката, която управлява твърдия диск и други устройства, са свързани помежду си посредством множество електрически линии, наречени **шина**

Хардуер и софтуер

- Съвкупността от електронните, електромеханичните и механичните компоненти, изграждащи компютъра, се нарича негова **апаратна част** или **хардуер**
- Хардуерът на компютъра е тясно свързан с т. нар. **програмна част**, **програмно осигуряване** или **софтуер**



Софтуер

- Програмното осигуряване на компютрите включва:
 - ☐ операционни системи;
 - ☐ среди за програмиране;
 - ☐ приложни програми.

Операционни системи (ОС)

- Всеки компютър работи под управлението на някаква ОС. Тя управлява:
 - взаимодействието между потребителите и хардуера (чрез команден език);
 - периферните устройства (чрез драйвери);
 - файловете (чрез файловата система).
- Широко разпространени ОС са MS DOS, MS WINDOWS, UNIX, LINUX.


Среди за програмиране (СП)

- Те автоматизират цялостния процес по създаването, транслирането, тестването, изпълнението, изменението и документирането на програмите.
- Средите за програмиране включват:
 - ☐ език за програмиране,
 - ☐ транслатор (компилатор или интерпретатор), свързващ редактор,
 - ☐ изпълнителна система,
 - ☐ система за проверка на програми,
 - ☐ система за поддържане на библиотеки и
 - ☐ текстови редактори.
- Тези компоненти са свързани в система чрез управляваща програма



Приложни програми (ПП)

- Те реализират някакво приложение:
 - ☐ икономическо,
 - ☐ проектантско,
 - ☐ инженерно,
 - ☐ счетоводно,
 - ☐ медицинско
 - ☐ и др.



Програми и програмиране.

Езици за програмиране

Програми. Автоматизиране на програмирането

- За да бъде изпълнена една или друга обработка над входните данни, трябва да бъде написана редица от инструкции (команди), които компютърът разбира и изпълнението на които да води до решаването на някаква задача. За различните задачи, тези редици са различни

Програми

- Редица от инструкции, водеща до решаване на определена задача, се нарича **програма**, а процесът на писане на програми - **програмиране**
- Инструкциите, с помощта на които се записва програма, изграждат език, наречен **език за програмиране**. Всеки компютър си има свой собствен език за програмиране (машинен език) и може да изпълнява програми, написани на него

Машинен език

- Машинният език е множество от машинни инструкции, които процесорът на компютъра може директно да изпълни. Програмата на машинен език е редица от числа, записани в двоична позиционна система.
- Една типична редица от машинни инструкции е следната:
 1. Премести стойността на клетка 15000 от паметта в регистър AX.
 2. Извади 10 от регистъра AX.
 3. Ако резултатът е положително число, изпълни командата, намираща се в клетка 25000 от паметта

Машинен език ...

- За различните процесори тази редица от инструкции се кодира по различен начин. При процесора Intel 80386 това кодиране има вида:

161 15000 45 10 127 25000

- Тази редица от числа се записва в паметта, след което се изпълнява. Но една дълга програма е съставена от хиляди команди и вероятността за грешки при програмиране на машинен език е много голяма

Асемблерни езици

- Първата стъпка в процеса на автоматизация на програмирането е въвеждането на **асемблерните езици**. Те дават кратки имена на командите. Например:
 - MOV означава премести,
 - SUB – извади,
 - JG – ако е по-голямо от 0, премини към

Асемблерни езици ...

- Като се използват тези команди, горната редица от инструкции се свежда до
MOV AX, [15000]
SUB AX, 10
JG 25000
- Този фрагмент е по-лесен за четене от хора, но компютърът не разбира тези инструкции. Те трябва да бъдат преведени на машинен език. Тази задача се изпълнява от компютърна програма, наречена **асемблер**.

Асемблерни езици ...

- Асемблерът взема редицата от букви MOV AX и я превежда в машинния код 161. Подобни действия асемблерът извършва и над другите команди. Асемблерът именува също и думите от паметта. Това позволява да бъдат използвани буквени имена вместо адреси от паметта. Например, горният фрагмент може да се запише и във вида:

```
MOV AX, [A]
```

```
SUB AX, 10
```

```
JG WRITE_ERROR
```

Езици за програмиране от високо НИВО

- В средата на 50-те години започнаха да се появяват езици за програмиране от високо ниво. При тях програмистът описва схематично основната идея за решаване на задачата, а специална програма, наречена **транслатор (компилатор или интерпретатор)**, превежда това описание в машинни инструкции за конкретния процесор. Например, на езика Паскал, горната редица от инструкции ще се запише във вида:

```
if a > 10 then writeln('error')
```

- Работата на компилатора е да обработи тази редица от букви и да я преведе във вида:
161 15000 45 10 127 25000

Езици за програмиране

Класификация

- Сега съществуват стотици различни езици за програмиране. Те могат да бъдат класифицирани по различни признаци. Най-общоприетата класификация е по степента на зависимост на езика от компютъра.

машинно-зависими

машинно-независими

машинни

асемблерни

макроезици

процедурно
ориентирани

проблемно
ориентирани

Класификация ...

- Машинно-независимите езици се наричат още **езици от високо ниво**.
- В настоящия курс ще направим въведение в програмирането на базата на езика за процедурно и обектно-ориентирано програмиране C++, който притежава и средства за програмиране почти на машинно ниво.

Класификация ...

- Друга класификация на езиците за програмиране е *според стила на програмиране*. Съществуват два основни стила за програмиране: **императивен (процедурен)** и **декларативен (дескриптивен)**.
- При процедурния стил програмата се реализира по схемата:

Програма = Алгоритъм

+

Структури от данни

Класификация ...

- Съществуват стотици процедурни езици. Такива са машинните и асемблерните езици, езиците Фортран, Алгол, Кобол, Снобол, Паскал, Ада, С, С++ и др.
- Процедурният стил е тясно свързан с фон Ноймановата архитектура на изчислителна система. Процедурните езици за програмиране могат да се разглеждат като синтактични обобщения на Ноймановите компютри. Семантиката на тези езици по същество е същата като семантиката на машинните езици. Следователно, процедурните езици са абстрактни версии от високо ниво на фон Ноймановите компютри и като такива, те наследяват всички техни недостатъци.

Класификация ...

- Декларативните езици са антипод на императивните. При тях в програмата се указва явно какви свойства има желаният резултат, без да се указва как точно този резултат се получава.
- Допустим е всеки начин за построение на програмата, който води до получаването на резултат с описаните в програмата свойства. Така при декларативните езици се казва какво се пресмята, без да се определя как да се пресмята.

Класификация ...

- Програмата на декларативен език за програмиране се определя така:

***Програма = Дефиниции на функции или
Равенства или
Правила + Факти***

- а изпълнението ѝ е **редукция, извод** или **доказателство**.
- Такива езици са Лисп, Пролог, Pop-11, TabLog, EqLog, Prolog-with-Equality и др.

Описание на език за програмиране

- За да се опише един език за програмиране е необходимо да се опишат неговите **синтаксис** и **семантика**.
- Да се определи **синтаксисът на един език за програмиране** означава да се зададе множество от правила, които определят как програмите се изграждат като редици от символи, т.е. да се зададе множество от правила, съгласно които една редица от символи е програма.
- Да се определи **семантиката на език за програмиране** означава да се определят правилата, съгласно които се изпълняват програмите на този език за програмиране.

Метаезик на Бекус-Наур

- Това е първото широко използвано формално означение за описание на синтаксиса на език за програмиране. Въведено е като механизъм за описание на синтаксиса на езика Algol 60.
- При това описание, имената на синтактичните категории се ограждат чрез ъглови скобки.
- *Пример*
<програма>, <число>, <цяло_число>

Метаезик на Бекус-Наур ...

- Знакът ::= означава “това е” или “дефинира се да бъде”. Синтактичните категории, които се дефинират, се поставят отляво на знака ::=:, а дефиницията им – отдясно на този знак.
- *Пример*
$$\langle \text{реално_число} \rangle ::= \langle \text{цяло_число} \rangle . \langle \text{цяло_число_без_знак} \rangle$$

Метаезик на Бекус-Наур ...

- Знакът | определя алтернативни конструкции и означава ИЛИ.
- *Пример*
- $\langle \text{цифра} \rangle ::= 0 \mid 1 \mid 2 \mid \dots \mid 9$

Метаезик на Бекус-Наур ...

- Означението {...} има смисъла, че ограденото в него се повтаря 0, 1 или повече пъти.
- *Пример*
 <редица_от_променливи> ::= <променлива>
 {, <променлива>}
- означава, че редица от променливи се дефинира като
 <променлива> или
 <променлива>, <променлива> или
 <променлива>, <променлива>, <променлива> и т.н.

Метаезик на Бекус-Наур ...

- Означението [...] има смисъла, че ограденото в скобите може да участва в описанието, но може и да бъде пропуснато.

- *Пример*

<цяло_число> ::=

[+ | -] <цяло_число_без_знак>

Алгоритъм

- При процедурния стил програмата се реализира по схемата:

Програма = Алгоритъм

+

Структури от данни

- Механизъм за намиране на решение, който е еднозначен, изпълним и завършващ, се нарича **алгоритъм**.
- Алгоритъмът като понятие е въведено от Евклид. Наречен е така в чест на арабския математик Ал Хорезми. В средновековието формулата DIXIT ALGORIZMI (така е казал Ал Хорезми) била сертификат за безупречна яснота и достоверност.

Алгоритъм ...

- *Еднозначността* гарантира точността на инструкциите на всяка стъпка и определя следващото действие.
- Свойството *изпълнимост* определя, че всяка стъпка може да се изпълни на практика,
- а *завършването* – че изпълнението на стъпките ще завърши.

Алгоритъм ...

- Основни начини за описание на алгоритъма са словесният, чрез блок-схема, чрез средствата на някакъв език за програмиране.
- *Пример.* Следващият фрагмент дава *общо* описание на алгоритъм за намиране на най-големия елемент на редицата от числа a_1, a_2, \dots, a_n .
 1. $m = a_1$
 2. Сравнява се a_2 с m и ако a_2 е по-голямо от m , то $m = a_2$
 3. Стъпка 2 се повтаря до изчерпване на редицата.

Алгоритъм ...

- Следва детайлно описание на алгоритъма за намиране на най-големия елемент на редицата от числа a_1, a_2, \dots, a_n .

1. $m = a_1$
2. $i = 2$
3. **Ако** $a_i > m$ **То** изпълни 4 **ИНАЧЕ** изпълни 5
4. $m = a_i$
5. $i = i + 1$
6. **Ако** $i \leq n$ **То** изпълни 3 **Иначе** изпълни 7
7. $\max = m$
8. съобщи \max
9. прекрати изпълнението

Алгоритъм ...

- При това описание n и a_1, a_2, \dots, a_n съдържат входните данни и се наричат още **входни променливи**, m съдържа резултата и се нарича **изходна променлива**, а t и i са **работни (вътрешни, междинни, помощни) променливи**.
- Решаването на една задача чрез програма на процедурен език се свежда до точно описание на това как да се получи резултатът.