Лабораторно упражнение № 2:

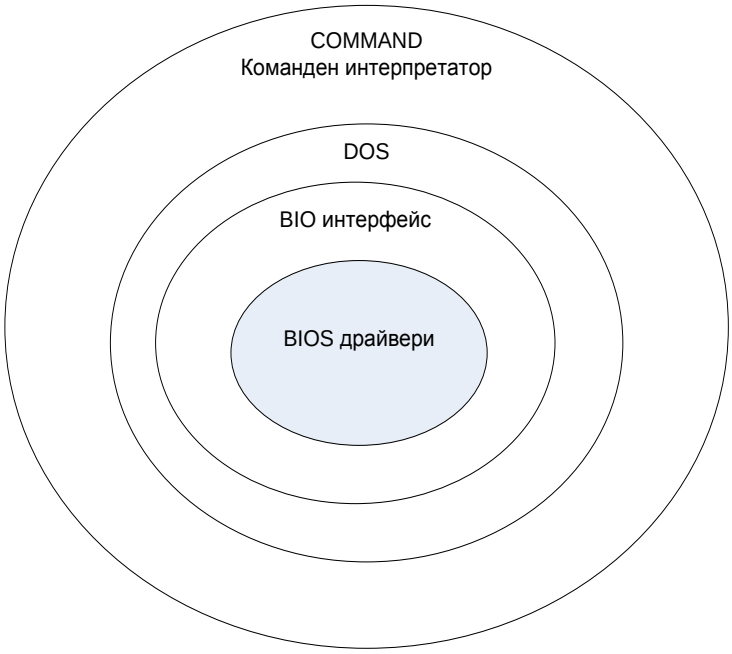
2. IDE - MS Visual Studio 2010. Алгоритми и начин за представянето им.Структура на С-програма. IDE - MS Visual Studio 2010

времетраене: 2 уч. часа

2.1 Основни компоненти на компютърна система. Запознаване с DOS. Файлови системи.

2.1.2 Дискови операционни системи – общо понятие.

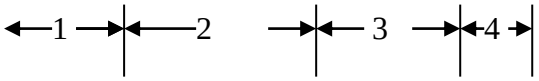
- MS DOS – едно програмна ОС



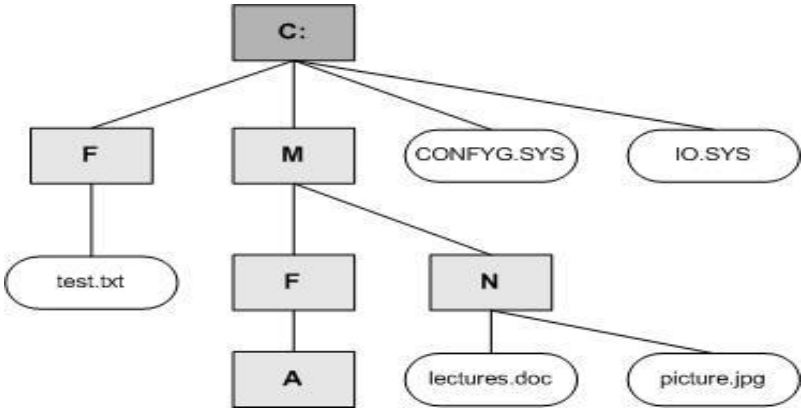
- Windows – многопрограмна ОС (Kernal – йерархична структура)

2.1.2 Файлова система.

- дискови носители – физическа и логическа структура – дървовидна структура при разпределение на дисковото пространство
- име на файл:

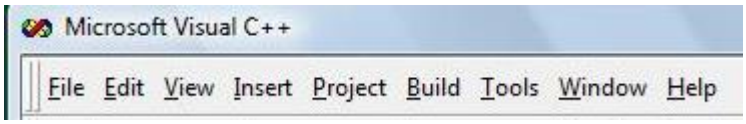


device : \path\....\path\ name_file . ext



2.2 Обща характеристика на програмната среда за проектиране на С- програмно осигуряване – MS Visual Studio 2010 – създаване на работен проект.

- Компоненти на IDE – редактор, компилатор, свързващ редактор, дебъгер.
- Основно меню на средата MS Visual Studio 2010 – функционално предназначение.



Понятие за работен проект – генериране на работен проект с помощта на Wizard на IDE,структура – работно пространство,**solution**

Последователност от действия за генериране на празен проект:

- от основното меню -> **File** –> **New** -> **Project .....**
- в диалоговия прозорец <**New Project**> се избира:
- от списъка <**Recent Templates**> се избира <**Visual C++**>, а в съседния списък <Empty Project>
- в полето за въвеждане <**Name**> се въвежда името на проекта – например: First (<Enter_name>)
- в полето <> се въвежда името на папката, в която се ще се съхраняват файловете на проекта (бутона **Browse**, позволява избора на папка за разполагане на проекта чрез графичен диалог – подобен на избора на файл с програмата File Explorer Windows) .
- в полето <**Solution name**> се въвежда име на проекта

След попълването на полетата се потвърждава създаването на проекта с бутона **ОК**.

При попълване на гореизброените полета от диалоговия прозорец автоматично се стартира модула <**MFC Application Wizard>,** който генерира новия проект, създавайки работно пространство с задължителните файлове за работа на **IDE Visual 2010.** Създава се папка със име еднакво с името на проекта, в която се разполагат работните му файлове.

При нормални настройки на средата, автоматично се отваря прикачения прозорец **Solution Explorer** , в който в дървовидна структура са извеждат генерираните папки и файлове. Обикновено средата създава три групи структури от файлове: **source files, header files, resource files**, които се разполагат в едноименната директория на проекта.

Следващата стъпка е добавянето или създаването на файлове, съдържащи кода на С-програмата:

- в прозореца на **Solution Explorer** се маркира с мишката **source files,** след което реално има две възможности – създаване на нов файл със изходния код на потребителската програма или добавяне на съществуващ (предварително създаден) файл. Действията са както следва:

- с десен бутон на мишката се извежда падащо меню за избор на нов компонент в проекта – избира се <**Add**>, появява се ново падащо меню, като при нов компонент се избира <**New Item…**>, а при добавяне на съществуващ <**Existing Item…**>.

При създаване на нов файл се попълва името на файла в диалоговия прозорец (задължително в името се попълва и разширението на файла **xxxxx.c**).

- Работа с редактора – меню File, Edit и View бързи клавиши за редакция на текста на С-програма (маркиране, копиране, изтриване, вмъкване, преместване ....).

2.3 Понятие за алгоритъм – свойства. Начини за представяне на алгоритмите.

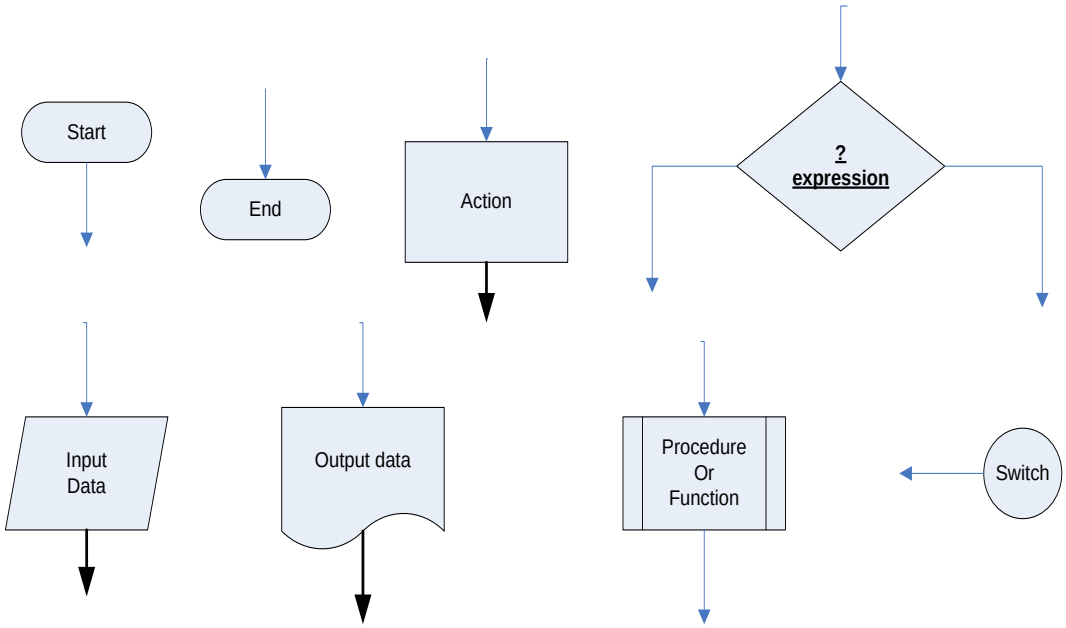
Понятие за алгоритъм – свойства. Начини за представяне на алгоритмите

определение: Формално описание на множеството от входни данни и последователността от действия, които се извършват над тях, с цел получаване на резултат, притежаващо следните свойства:

- дискретност
- определеност
- масовост
- резултантност

2.4 Начини за записване на алгоритмите.

- текстово
- блоков алгоритъм



с език от високо ниво

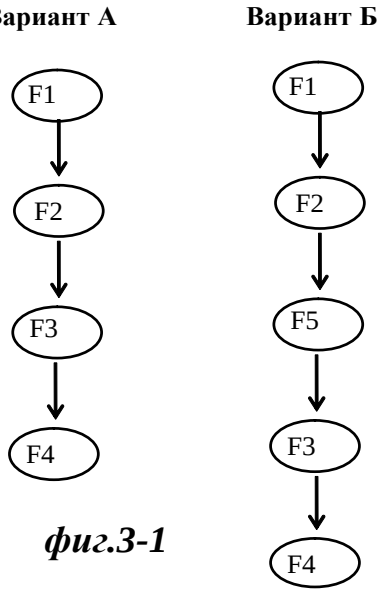
2.5 Структура на С-програма.

пример P_1-1 ⇨

Source (изходен) текст на С програма

вариант А⇩

```
/*  Програма за извеждане на по-голямото по
стойност число от  входна двойка цели числа
*/

/* ------------ начало на фрагмент F1  -------------------
------------*/
#include <Visual_2010.h>
#include <stdio.h>
#include <conio.h>
/*  ----------- край на фрагмент F1  ---------------------------------*/


/* ----------- начало на фрагмент F2  ---------------------------------*/
int a1,a2;
/*  ----------- край на фрагмент F2  ---------------------------------*/

/* ----------- начало на фрагмент F3  ---------------------------------*/
int max_value( int a, int b) {

int max;    /* локална променлива на функцията */
/*--- операторна част на функцията  */
        max=a;
        if (max<b) max=b;
        return max;
}
/*  ----------- край на фрагмент F3  ---------------------------------*/
/* ----------- начало на фрагмент F4  ---------------------------------*/

int main (void ) {
int a,b,s;

/*--- операторна част на главната функцията  */

        printf (”\nвъведете стойност за a1=”);
        scanf(”%d”,&a1);
        printf (”\nвъведете стойност за a2=”);
        scanf(”%d”,&a2);
        s=max_value(a1,a2);
        printf(”\n\nот двете стойности %d и %d по-голямата е %d”,a1,a2,s);
        printf (”\nвъведете стойност за a=”);
        scanf(”%d”,&a);
        printf (”\nвъведете стойност за b=”);
        scanf(”%d”,&b);
        s=max_value(a,b);
        printf(”\n\nот двете стойности %d и %d по-голямата е %d”,a,b,s);
       return 0;
}
/*  ----------- край на фрагмент F4  ---------------------------------*/
```



фиг.3-1

 вариант Б⇩

```
/*  Програма за извеждане на по-голямия по стойност елемент */
/*  от входна двойка цели числа    - вариант Б на source кода      */

/* ------------ начало на фрагмент F1  ---------------------------------*/

#include <Visual_2010.h>
#include <stdio.h>
#include <conio.h>

/*  ------------ край на фрагмент F1  ---------------------------------*/

/* ------------ начало на фрагмент F2  ---------------------------------*/
int a1,a2;
/*  ------------ край на фрагмент F2  ---------------------------------*/
/* ------------ начало на фрагмент F5  ---------------------------------*/
int max_value(int a, int b);
/*  ------------ край на фрагмент F5  ---------------------------------*/

/* ------------ начало на фрагмент F4  ---------------------------------*/
int main (void ) {

int a,b,s;
/*--- операторна част на главната функция  */
        printf (”\nвъведете стойност за a1=”);
        scanf(”%d”,&a1);
        printf (”\nвъведете стойност за a2=”);
        scanf(”%d”,&a2);
        s=max_value(a1,a2);
        printf(”\n\nот двете стойности %d и %d по-голямата е %d”,a1,a2,s);
        printf (”\nвъведете стойност за a=”);
        scanf(”%d”,&a);
        printf (”\nвъведете стойност за b=”);
        scanf(”%d”,&b);
        s=max_value(a,b);
        printf(”\n\nот двете стойности %d и %d по-голямата е %d”,a,b,s);
      return 0;
}
/*  ------------ край на фрагмент F4  ---------------------------------*/

/* ------------ начало на фрагмент F3  ---------------------------------*/
int max_value( int a, int b) {

int max;    /* локална променлива на функцията */
/*--- операторна част на функцията  */
        max=a;
        if (max<b) max=b;
        return max;
}
/*  ------------ край на фрагмент F3  ---------------------------------*/
```