

Управляващи структури

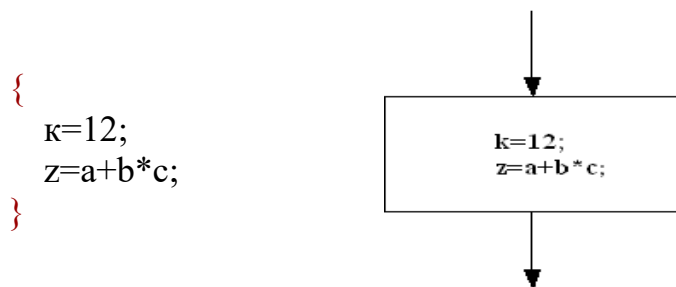
Всички управляващи структури имат един вход!

Управляващите структури служат за

- избор на отделни секции от програмата
- повторение на отделни секции от програмата - цикли

Могат да се включват една в друга. Не могат да се пресичат (имат един вход)

1) блок - един вход и един изход



2) разклонение

if (логически израз) блок

Пример:

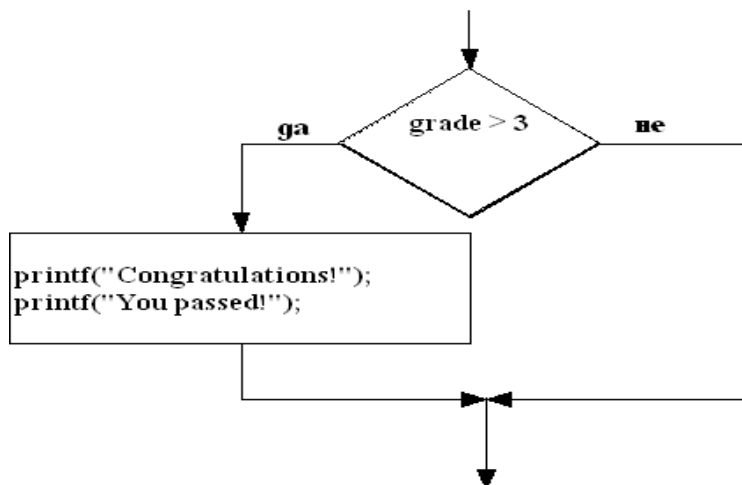
```
int grade = 4;
if( grade > 3 )
    printf("Congratulations!");
```

Пример:

```
int grade = 4;
if( grade > 3 ) {
    printf("Congratulations!");
}
```

Пример:

```
int grade = 4;
if( grade > 3 ) {
    printf("Congratulations!");
    printf("You passed!");
}
```



if (логически израз) блок else блок

Пример:

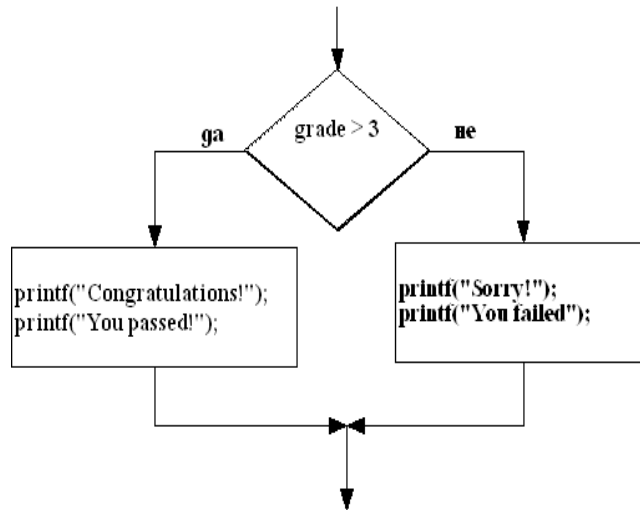
```
int grade = 4;
if( grade > 3 )
    printf("Congratulations!");
else
    printf("Sorry!");
```

Пример:

```
int grade = 4;
if( grade > 3 ) {
    printf("Congratulations!");
}
else {
    printf("Sorry!");
}
```

Пример:

```
int grade = 4;
if( grade > 3 ) {
    printf("Congratulations!");
    printf("You passed");
}
else {
    printf("Sorry!");
    printf("You failed");
}
```



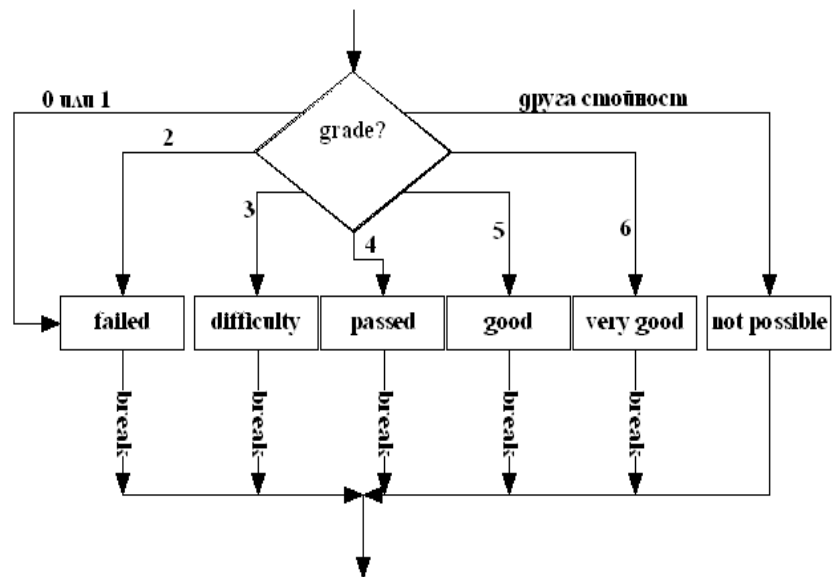
3) многозначно разклонение

```
switch ( целочислен израз ) {
    case константен израз : инструкции
    ...
    case константен израз : инструкции
}
```

```
switch ( целочислен израз ) {
    case константен израз : инструкции
    ...
    case константен израз : инструкции
    default: инструкции
}
```

Пример:

```
int grade = 4;
switch (grade) {
    case 0:
    case 1:
    case 2: printf("You failed!"); break;
    case 3: printf("with great difficulty!");
    break;
    case 4: printf("You passed!"); break;
    case 5: printf("You are good");break;
    case 6: printf("You are very good");break;
    default: printf("This is not possible!");
}
```

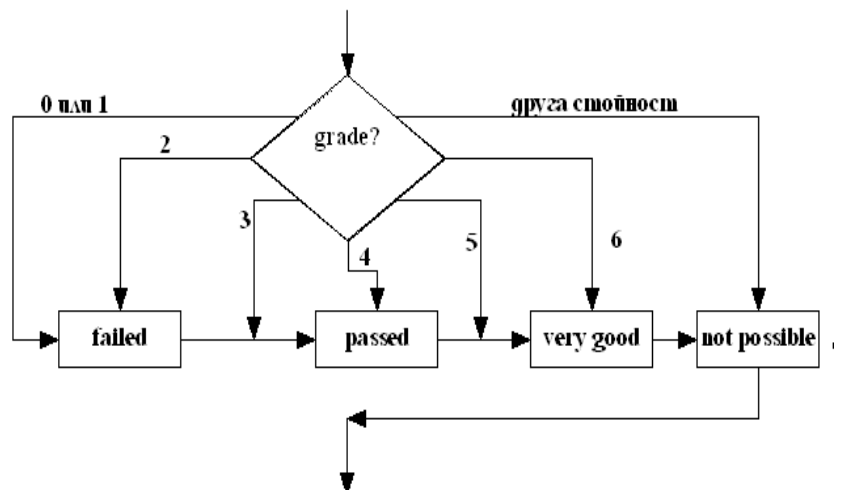


break; Прекратява изпълнението на управляващата структура

Начинът на подреждане на *case* и *default* е от съществено значение за резултата от изпълнението.

Пример:

```
int grade = 4;
switch (grade) {
    case 0:
    case 1:
    case 2: printf("You failed!");
    case 3:
    case 4: printf("You passed!");
    case 5:
    case 6: printf("You are very
    good");
    default: printf("This is not
    possible!");
}
```

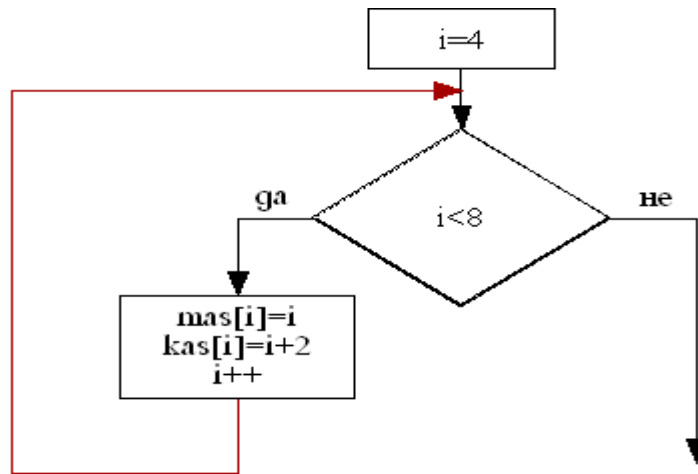


4) Цикъл с предусловие - while

while(логически израз)
блок

Пример:

```
i=4;  
while(i<8){  
    mas[i]=i;  
    kas[i]=i+2;  
    i++;  
}
```



6) Цикъл с предусловие - for

for(израз 1; логически израз 2; израз 3) блок

израз 1 - подготовка за цикъла (вкарана е в конструкцията за разлика от while цикъл)

логически израз 2 - проверка за край на цикъла

израз 3 - подготовка за следваща стъпка (итерация)

блок - тяло на цикъла (повтарящите се действия)

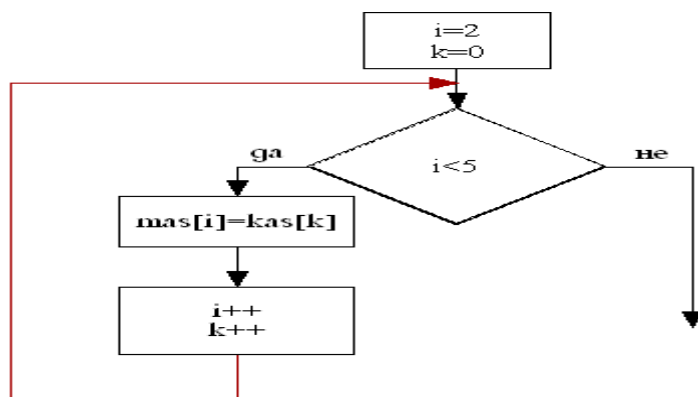
обърнете внимание на ';' - трябва да има задължително 2!

ред на изпълнение (изпълнява се само ако условието е истина):

- 1) израз 1
- 2) логически израз 2; ако резултатът не е истина – към край
- 3) блок
- 4) израз 3
- 5) към стъпка 2

Пример:

```
for(i=2,k=0;i<5;i++,k++){  
    mas[i]=kas[k];  
}
```



8) Цикъл с пост-условие do-while

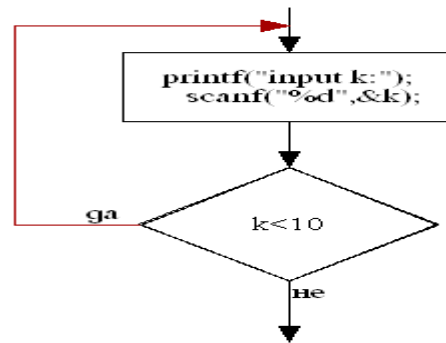
```
do  
    block  
while(логически израз)
```

Ред на изпълнение (изпълнява се поне един път):

- 1) block
- 2) логически израз, ако е истина - към 1

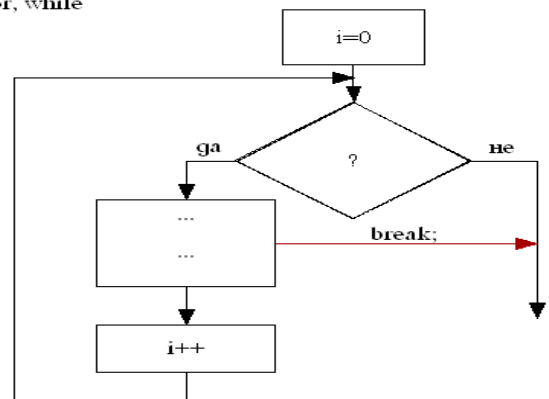
пример:

```
do{  
    printf("input k:");  
    scanf("%d",&k);  
} while (k<10)
```



9) break - Прекратява изпълнението на управляващата структура. В цикъл се изпълнява винаги при наличието на някакво условие налагащо прекратяването на цикъла. Еднакво действие при всички цикли – while, for и do-while.

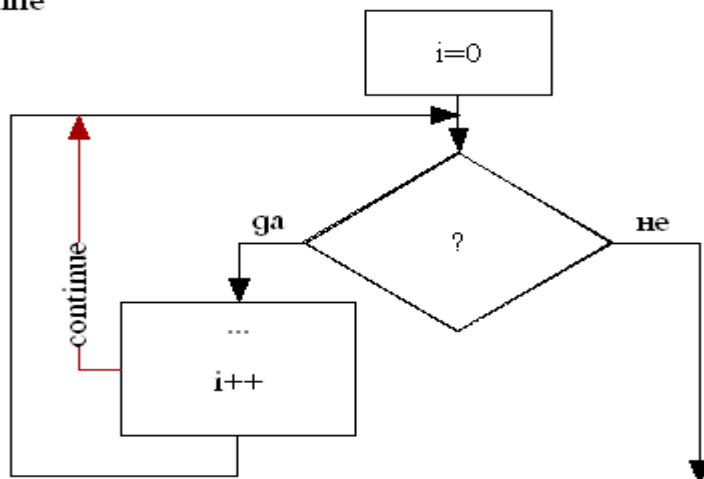
for, while



10) continue :

- В while и в do-while цикъл, се преминава към условието за край на цикъл.

while

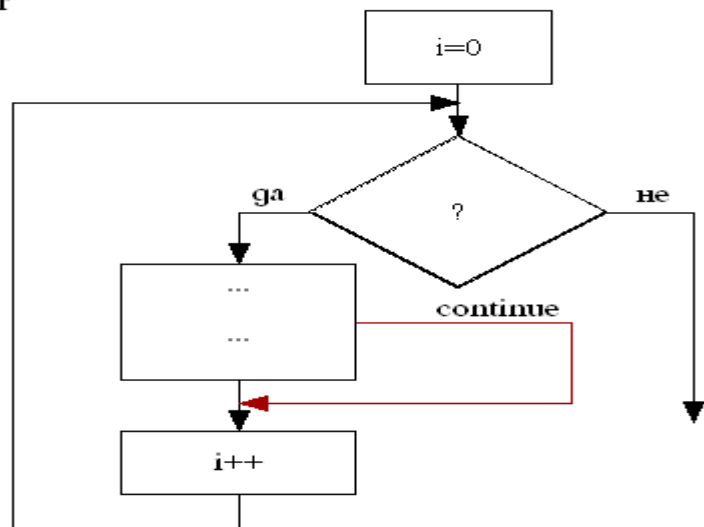


```
#include <stdio.h>
int main(){
    int i=0;
    while(i<7){
        if(i%2) continue;
        printf("%d ",i);
        i++;
    }
}
```

Извежда на екрана 0 и след това зацикля на i със стойност 1

- В for цикъл, се преминава към подготовката за следваща стъпка

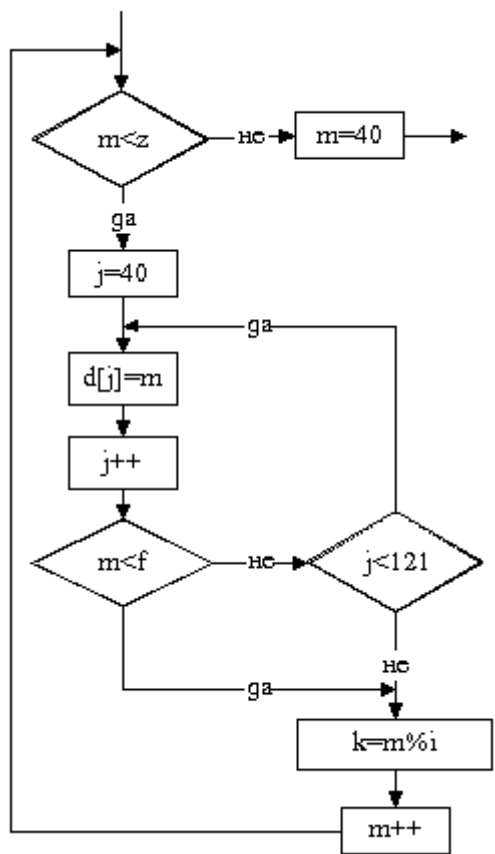
for



```
#include <stdio.h>
int main(){
    int i=0;
    for(i=0;i<7;i++){
        if(i%2) continue;
        printf("%d ",i);
    }
}
```

извежда на екрана
0 2 4 6

11. Пример: част от програма с няколко структури



```
for( ; m<z;m++) {  
    j=40;  
    do {  
        d[j]=m;  
        j++;  
        if(m<f) break;  
        while(j<121);  
        k=m%i;  
    }  
    m=40;  
}
```