

24.3.2011

## Изброим тип

enum <име> { <СПИСЪК ОТ СТОЙНОСТИ> } ;

пример:

```
enum Cardsuit { Club, Diamond, Heart, Spade };
int main()
{
    enum Cardsuit card1,card2;
    card1 = Heart; card2=Spade;
    printf("card2:%d\n",card2);
    printf("card2-card1:%d\n",card2-card1);
}
```

## Бинарни файлове

rb, r+b, wb, w+b, ab, a+b

```
int fgetc( FILE *stream ); прочетен СИМВОЛ/EOF
int fputc( int ch, FILE *stream ); ch/ EOF
int fread( void *buffer, size_t size, size_t num, FILE *stream );
int fwrite( const void *buffer, size_t size, size_t count, FILE *stream );
```

```
int fseek(FILE *stream, long offset, int whence);
```

0-ok, SEEK\_SET, SEEK\_CURR, SEEK\_END

```
long ftell(FILE *stream); -1 on error
```

```
void rewind(FILE *stream);
```

пример – разлика при четене от текстов и бинарен

```
#include <stdio.h>
int main(int argc, char *argv[]){
    int i;
    char ms[]= "abraka\nабрака";
    FILE *ft,*fb;
    if(!(ft=fopen("tst.dat","wt"))) /* ...*/;
    for(i=0;ms[i];i++){
        if(fputc(ms[i],ft)==EOF) /*...*/;
    }
    fclose(ft);
    printf ("\n\n text:");
    if(!(ft=fopen("tst.dat","rt")))
        /* ...*/;
    for(int i=0;!feof(ft);i++){
        if(!(i%9))printf("\n");
        printf("%d\t", fgetc(ft));
    }
    rewind(ft);
    printf ("\n\n text 2:");
    for(int i=0;!feof(ft);i++){
        char c;
        if(!(i%9))printf("\n");
        if(fscanf(ft,"%c",&c)==-1) printf(" EOF"); else printf("%d\t", c);
    }
    rewind(ft);
    printf ("\n\n text 3:");
    for(int i=0;!feof(ft);i++){
        unsigned char c;
        if(!(i%9))printf("\n");
        printf("%d\t", fgetc(fb));
    }
    fclose(fb);
    return 0;
}
```

```
text:
97 98 114 97 107 97 10 224 225
240 224 234 224 -1
text 2:
97 98 114 97 107 97 10 -32 -31
-16 -32 -22 -32 EOF
text 3:
97 98 114 97 107 97 10 224 225
240 224 234 224 EOF
binary:
97 98 114 97 107 97 13 10 224
225 240 224 234 224 -1
```

пример

```
#include <stdio.h>
typedef enum {ugly, good_1 } look;
typedef enum {evil,good} temper;
typedef enum {jan,feb,mars,apr,may} month;
typedef struct {
    int day;
    month mnth;
    int year;
} date;
typedef struct {
    char name[50];
    date bd;
    temper tmpr;
    look lk;
} person;

int func(FILE *);
int main(int argc, char *argv[]){
    int i;
    FILE *f;
    if(!(f=fopen("tst.dat","w+b")))return 1;
    person *pp,p[20]={
        {"Ani",{21,jan,1989},evil,good_1},
        {"Kira",{10,feb,1986},good,good_1},
        {"Sima",{11,mars,1985},evil,ugly}
    };
    /* ... */

    for(i=0; i<20; i++){
        if(!fwrite(p+i,sizeof(person),1,f)){ fclose(f);return 2;}
        if( (p[i].lk == good_1)
            printf("\n found good looking: %s",p[i].name);
    }
    for(pp=p;pp-p<20;pp++){
        if ((pp->tmpr == good) &&
            (pp->lk == good_1) &&
            (pp->bd.mnth == feb) )
            printf("\n found good temper, good_look, in feb bday:%s\n",pp->name);
    }
    func(f); fclose(f);
    return 0;
}

int func(FILE * f){
    person p;
    fseek(f, 2*sizeof(p),SEEK_SET);
    if(!fread(&p,sizeof(p),1,f)return 5;
    printf("\n the third person is %s\n",p.name); return 0;
}
```

```
found good looking: Ani
found good looking: Kira
found good temper, good_look, in feb bday: Kira
```

```
the third person is Sima
```