

5. Studienarbeit

1. Version: Signatur mit einzelner Pixelbelegung

```

#include<graphics.h>
#include<conio.h>
#include<stdlib.h>
#include<stdio.h>
#define xwert 600 /*Konstanten*/
#define ywert 400
#define DIM 10

relem(int x_pos,int y_pos); /*Funktionsprototyp*/

main()
{
    int graphdriver, /*Grafiktreiber*/
        graphmode, /*Grafikmodus*/
        errorcode, /*Für Fehlermeldung*/
        i,xpos,ypos;

    graphdriver=DETECT; /*Setzt den Treiber automatisch*/

    initgraph(&graphdriver,&graphmode,"W:\\BC.31\\BGI"); /*Start des Grafikpakets*/
    errorcode=graphresult(); /*Fehlerüberprüfung*/
    if (errorcode!=grOk) /*Treiber gefunden ?*/
    {
        puts(grapherrormsg(errorcode)); /*Fehlermeldung*/
        exit(1);
    }

    for (i=1;i<=DIM;i++) /*Erstellen von 10 Stk. an zufälligen Positionen*/
    {
        xpos=random(xwert);
        ypos=random(ywert);
        relem(xpos,ypos);
    }
    getch();
    closegraph(); /*Ende Grafikpaket, ab hier wieder alter (Text-)Modus*/
}
relem(int xpos,int ypos) /*Funktion für die Pixelbelegung der Signatur*/
{
    putpixel(xpos+4,ypos+0,14); /*Figur*/
    putpixel(xpos+12,ypos+0,14);
    putpixel(xpos+3,ypos+1,14);
    putpixel(xpos+4,ypos+1,14);
    putpixel(xpos+5,ypos+1,14);
    putpixel(xpos+11,ypos+1,14);
    putpixel(xpos+12,ypos+1,14);
    putpixel(xpos+13,ypos+1,14);
    putpixel(xpos+2,ypos+2,14);
    putpixel(xpos+3,ypos+2,1);
    putpixel(xpos+4,ypos+2,14);
    putpixel(xpos+5,ypos+2,1);
    putpixel(xpos+6,ypos+2,14);
    putpixel(xpos+10,ypos+2,14);
    putpixel(xpos+11,ypos+2,1);
    putpixel(xpos+12,ypos+2,14);
    putpixel(xpos+13,ypos+2,1);
}

```

```
putpixel(xpos+14,ypos+2,14);
putpixel(xpos+3,ypos+3,14);
putpixel(xpos+4,ypos+3,14);
putpixel(xpos+5,ypos+3,14);
putpixel(xpos+11,ypos+3,14);
putpixel(xpos+12,ypos+3,14);
putpixel(xpos+13,ypos+3,14);
putpixel(xpos+0,ypos+4,14);
putpixel(xpos+8,ypos+4,14);
putpixel(xpos+16,ypos+4,14);
putpixel(xpos+4,ypos+4,14);
putpixel(xpos+12,ypos+4,14);
putpixel(xpos+1,ypos+5,4);
putpixel(xpos+2,ypos+5,4);
putpixel(xpos+3,ypos+5,4);
putpixel(xpos+4,ypos+5,4);
putpixel(xpos+5,ypos+5,4);
putpixel(xpos+6,ypos+5,4);
putpixel(xpos+7,ypos+5,4);
putpixel(xpos+9,ypos+5,2);
putpixel(xpos+10,ypos+5,2);
putpixel(xpos+11,ypos+5,2);
putpixel(xpos+12,ypos+5,2);
putpixel(xpos+13,ypos+5,2);
putpixel(xpos+14,ypos+5,2);
putpixel(xpos+15,ypos+5,2);
putpixel(xpos+3,ypos+6,4);
putpixel(xpos+4,ypos+6,4);
putpixel(xpos+5,ypos+6,4);
putpixel(xpos+11,ypos+6,2);
putpixel(xpos+12,ypos+6,2);
putpixel(xpos+13,ypos+6,2);
putpixel(xpos+3,ypos+7,4);
putpixel(xpos+4,ypos+7,4);
putpixel(xpos+5,ypos+7,4);
putpixel(xpos+11,ypos+7,2);
putpixel(xpos+12,ypos+7,2);
putpixel(xpos+13,ypos+7,2);
putpixel(xpos+3,ypos+8,4);
putpixel(xpos+5,ypos+8,4);
putpixel(xpos+10,ypos+8,2);
putpixel(xpos+11,ypos+8,2);
putpixel(xpos+12,ypos+8,2);
putpixel(xpos+13,ypos+8,2);
putpixel(xpos+14,ypos+8,2);
putpixel(xpos+3,ypos+9,4);
putpixel(xpos+5,ypos+9,4);
putpixel(xpos+10,ypos+9,2);
putpixel(xpos+11,ypos+9,2);
putpixel(xpos+12,ypos+9,2);
putpixel(xpos+13,ypos+9,2);
putpixel(xpos+14,ypos+9,2);
putpixel(xpos+3,ypos+10,4);
putpixel(xpos+5,ypos+10,4);
putpixel(xpos+9,ypos+10,4);
putpixel(xpos+10,ypos+10,2);
putpixel(xpos+11,ypos+10,2);
putpixel(xpos+12,ypos+10,2);
putpixel(xpos+13,ypos+10,2);
putpixel(xpos+14,ypos+10,2);
putpixel(xpos+15,ypos+10,2);
putpixel(xpos+2,ypos+11,6);
putpixel(xpos+3,ypos+11,6);
putpixel(xpos+5,ypos+11,6);
putpixel(xpos+6,ypos+11,6);
putpixel(xpos+10,ypos+11,6);
putpixel(xpos+11,ypos+11,6);
putpixel(xpos+13,ypos+11,6);
putpixel(xpos+14,ypos+11,6);
}
```

5. Studienarbeit

2. Version: Signatur mit Pixelbelegung in Form eines Feldes

```

#include<graphics.h>
#include<conio.h>
#include<stdlib.h>
#include<stdio.h>
#define DIM 10                                /*Konstanten*/
#define xwert 300
#define ywert 200
#define breite 17                             /*Farbzuweisung*/
#define länge 12

char const bgi[]="W:\\BC.31\\BGI";
/*Funktion zum Lesen des Feldes*/
void relem (int Farbe[länge][breite] ,int xpos, int ypos)
{
    int i,j;
    for (i=0;i<=breite;i++)                  /*Durchlauf des Feldes*/
        for (j=0;j<=länge;j++)
            putpixel (xpos+i,ypos+j,Farbe[j][i]);    /*Setzen der Farbe*/
}

void main()
{
    int graphdriver,                          /*Grafiktreiber*/
        graphmode,                            /*Grafikmodus*/
        errorcode,                            /*Für Fehlermeldung*/
        xpos,ypos,i,j;

    /*Signatur als Feld*/
    int Farbe [länge][breite]= {0,0,0,0,14,0,0,0,0,0,0,0,14,0,0,0,0,
                                0,0,0,14,14,14,0,0,0,0,0,14,14,14,0,0,0,
                                0,0,14,1,14,1,14,0,0,0,14,1,14,1,14,0,0,
                                0,0,0,14,14,14,0,0,0,0,0,14,14,14,0,0,0,
                                14,0,0,0,14,0,0,0,14,0,0,0,14,0,0,0,14,
                                0,4,4,4,4,4,4,4,0,4,4,4,4,4,4,4,4,0,
                                0,0,0,4,4,4,0,0,0,0,0,4,4,4,0,0,0,
                                0,0,0,4,4,4,0,0,0,0,0,4,4,4,0,0,0,
                                0,0,0,4,0,4,0,0,0,0,4,4,4,4,4,0,0,
                                0,0,0,4,0,4,0,0,0,0,4,4,4,4,4,0,0,
                                0,0,0,4,0,4,0,0,0,0,4,4,4,4,4,4,0,
                                0,0,4,4,0,4,4,0,0,0,4,4,4,4,4,4,4,0,
                                0,0,4,4,0,4,4,0,0,0,4,4,0,4,4,4,0,0};

    graphdriver=DETECT;                       /*Setzt den Treiber automatisch*/
    initgraph(&graphdriver,&graphmode,bgi);
    errorcode=graphresult();                  /*Start des Graphikpakets*/
    if (errorcode!=grOk)                     /*Fehlerüberprüfung*/
    {
        puts (grapherrormsg(errorcode));     /*Fehlermeldung*/
        exit(1);
    }

    for (i=1;i<=DIM;i++)                      /*Anzahl der Signaturen*/
    {
        xpos=random(xwert;
        ypos=random(ywert);
        relem(Farbe,xpos,ypos);              /*Funktionsaufruf*/
    }
    getch();                                  /*Ende Grafikpaket, ab hier wieder alter (Text-)Modus*/
    closegraph();                             /*Ende main*/
}

```