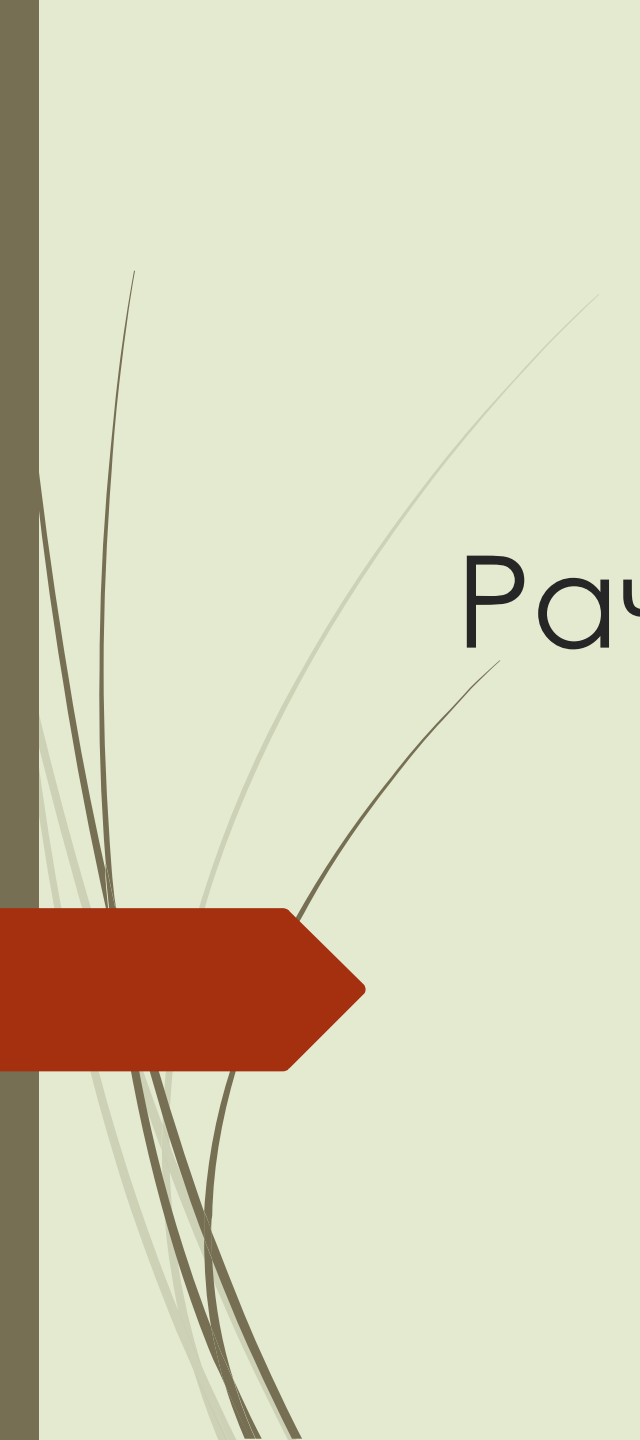




# Рачунарство и информатика

Јелица Васиљевић

Одељење II см

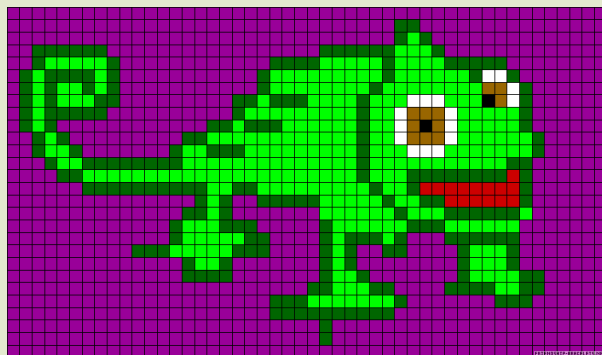
Школска 2015/2016

# Модул GRAPH

- Модул CRT омогућава графички рад али на врло ниском нивоу
  - Слика је формирана од карактера – слова, бројеви, специјални знаци
  - Није могуће дефинисати ( нацртати ) неки специфичан облик јер је јединица управљања **карактер**
- Модул **GRAPH** садржи функције и процедуре за управљање изгледом излазног екрана у графичком режиму рада
- У графичком режиму рада, екран је подељен на **пикселе**
  - Сваки пиксел представља једну осветљену тачку екрана
    - Прецизнија манипулација с обзиром на то да је пиксел далеко мањи од карактера
    - Димензије радне површине (прозора) су веће него у текстуланом режиму
  - Размаци између пиксела су довољно мали тако да су „празнине“ невидљиве људском оку – несавршеност ока

# Модул GRAPH

- Графички режим рада
- Курсор није видљив
- Број пиксела зависи од резолуције



# Модул GRAPH

- Текстуални режим рада је подразумевани. Да би се радило у графичком режиму рада, потребно је **експлицитно** позвати рутине ( процедуре или функције ) које ће извршити иницијализацију графичког окружења.
- Покретање графичког режима рада неопходно је урадити на почетку програма или непосредно пре коришћења графичких рутина.

■ Позивање процедуре **InitGraph (var GraphDriver, GraphModus : integer; const PathToDriver : string);**

- **GraphDriver** – Графички адаптер . Обично се ова променљива поставља на *Detect* вредност и тада аутоматски открива и поставља графички адаптер и режим рада
- **GraphModus** - Графички режим
- **PathToDriver** - Директоријум са одговарајућим драјвером

■ **Пример иницијализације :**

```
graphMode, GraphDriver: smallint;  
.....  
GraphDriver:=Detect;  
InitGraph (GraphDriver, graphMode, '');
```

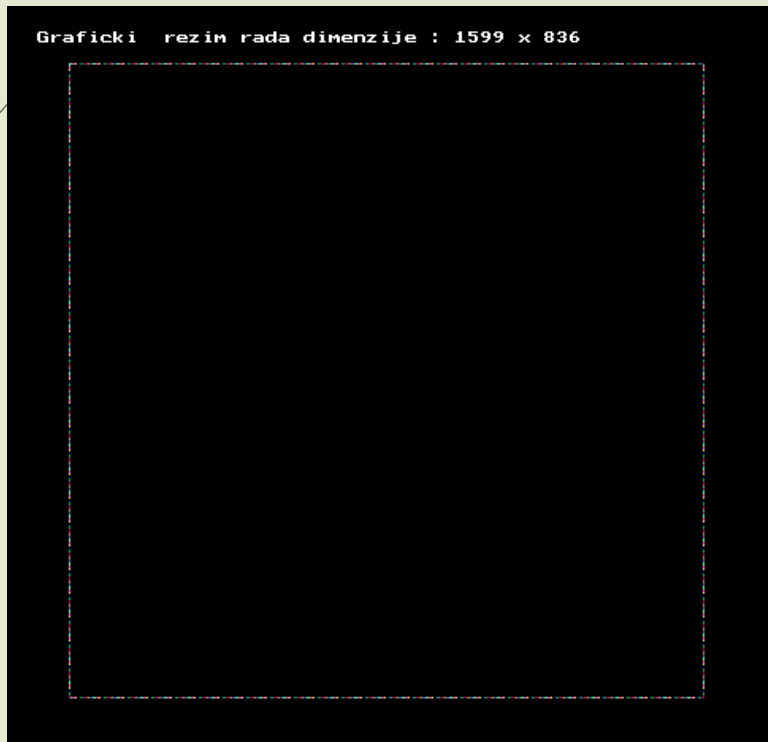
- Позивање процедуре **CloseGraph** брише се садржај екрана, адаптер се враћа у текстуални режим рада и ослобађа се меморија заузета учитаним драјвером.

# Модул GRAPH

- Након иницијализације графичког режима, могуће је позивати функције Graph модула.
- Неке од често коришћених функција су :
  - **OutText(text)** – Исписује текст на екрен у графичком режиму. Није могуће користити write/writeln у графичком режиму. То су процедуре текстуланог режима
  - **SetColor (boja)** - Постављање боје
  - **GetMaxX** - функција која враћа максималну вредност за хоризонталну координату
  - **GetMaxY** - функција која враћа максималну вредност за вертикалну координату
  - **ClearDevice** – Брише садржај екрана ( тренутном бојом ) и поставља курсор на позицију (0,0)
  - **MoveTo(x,y)** - Помера курсор на тачку са координатама (x,y)
  - **Rectangle(x1,y1,x2,y2)** – Исцртава правоугаоник чији горњи леви угао има координате (x1,y1) а доњи десни (x2,y2)
  - **Circle (x,y,r)** – исцртава круг са центром у (x,y) и полупречником r
  - **PutPixel(x,y,boja)** - Исцртава тачку на позицији (x,y) бојом boja
  - **Line(x1,y1,x2,y2)** – Исцртава линију од тачке са координатама (x1,y1) до тачке са координатама (x2,y2), тренутно подешеном бојом
  - **LineRel(x,y)** - Исцртава линију од **тренутне позиције курсора** до тачке са која је од њега удаљена x тачака по x-оси и y тачака по y-оси
  - **LineTo(x,y)** – Исцртава линију од **тренутне позиције курсора** до тачке са координатама (x,y)

# Модул GRAPH

- Написати програм који у графичком режиму
  - Исписује резолуцију екрана почевши од позиције (80,80).
  - Исцртава квадрат чији горњи леви угао има координате (100,100) и дужину странице 400. Сваки пиксел странице исцртати различитом бојом.



# Модул GRAPH



```
program Inicijalizacija;
uses Graph;
var
  graphMode, GraphDriver: smallint;
  i, xmax, ymax: integer;
  s, xs, ys: string;
begin
  GraphDriver:=Detect;
  InitGraph(GraphDriver,graphMode,'');
  xmax:=GetMaxX();
  ymax:=GetMaxY();
  str(xmax,xs);
  str(ymax,ys);
  MoveTo(80,80);
  outText('Graficki rezim rada dimenzije : '+xs+' x ' +ys);
  for i:= 100 to 500 do
    begin
      PutPixel(100,i,i mod 15 );
      PutPixel(i,100, i mod 15);
      PutPixel(500,i,i mod 15 );
      PutPixel(i,500, i mod 15);
    end;
    readln;
  Closegraph();
end.
```