

Struktura podataka koja na sintaksnom nivou uključuje i operacije nad tim podacima je klasa. Klasa je skup osobina objekta, a objekti su konkretni primeri, primerci, klasa. Klasa je novi tip podatka. Glavni program se piše tek kada se uspostavi dovoljan broj klasa i njihovih konkretizacija, tj. objekata.

Objekti su članovi klasa; klasa je porodica objekata, a objekti stvoreni tokom izvršenja programa su primerci klase. Klasa sadži metode i promenljive. Metode su procedure ili funkcije koje se pozivaju primanjem poruke iz spoljnog sveta. Klasa može sadržati privremene promenljive (koje gube vrednost pozavršetku metoda) i promenljive na nivou primerka (koje očitavaju individualna stanja bez obzira na izvršenje metoda).

Jedinice (juniti) su osnova modularnog programiranja u Turbo Pascal-u. preporučljivo je da svi metodi jedne klase budu u jednoj jedinici. Jedinica počinje zaglavljem, koje se sastoji iz službene reči **unit** i imena jedinice. Ime jedinice koristi se u naredbi **uses**. Unit se sastoji iz tri dela i svaki počinje određenom rezervisanom rečju: **interface**, **implementation** i **begin**. Ova poslednja se može izostaviti.

```
UNIT <ime unita>;
```

```
INTERFACE
```

Deklarišu se konstante, tipovi, varijable, procedure i funkcije javnog karaktera. Reč »javni« ovde znači da se svi elementi definisani kao interface mogu koristiti i u glavnom programu i u nekim drugim unitima. U ovom delu se navode i deklaracije procedura i funkcija.

```
IMPLEMENTATION
```

U njemu se nalaze tela procedura i funkcija deklariranih u interface delu. Sve što je definisano u ovom delu ostaje privatno za dati unit.

```
END.
```

Smisao podele unita na javne i privatne podatke i potprograme je tzv. prikrivanje informacija. Idealno bi bilo da u jednom unitu postoje samo operacije nad jednom strukturom.

### Primer1: Unit

```
unit j1;
```

```
interface
```

```
procedure SetVrednost(nova:integer);
```

```
function GetVrednost:integer;
```

```
implementation
```

```
var Vrednost:integer;
```

```
procedure SetVrednost(nova:integer);
```

```
begin
```

```
  Vrednost:=nova;
```

```
end;
```

```
function GetVrednost:integer;
```

```
begin
```

```
  GetVrednost:=Vrednost;
```

```
end;
```

```
end.
```

```
program tp1;
```

```
uses j1;
```

```
begin
```

```
  SetVrednost(5);
```

```
  writeln('Postavljena vrednost je: ',
```

```
    GetVrednost);
```

```
  readln
```

```
end.
```

Glavni program TP1 nema načina da dopre do promenljive Vrednost, jer je ona definisana u implementation delu. Glavni program postavlja apstraktnu vrednost pozivom procedure SetVrednost i očitava funkcijom GetVrednost. Zato se kaže da unit J1 definiše apstraktni tip. Zaista, unutrašnjost unita J1 možemo menjati, a da glavni program nastavi da i dalje radi bez izmena. Dakle, deklaracije u implementation delu omogućuju da se ostvari apstrakcija podataka.

## Primer2. Unit

Definišimo slog Pravougaonik i odgovarajuće operacije nad njim. Pravougaonik se sastoji iz dve celobrojne promenljive: x i y.

### unit j2:

```
interface
type  Pravougaonik=record
        x,y:integer;
    end;
procedure SetX(novo:integer);
function GetX:integer;
procedure SetY(novo:integer);
function GetY:integer;
procedure Prikaz;
procedure GetPravougaonik(var pr:Pravougaonik);
implementation
var p:Pravougaonik;
procedure SetX;
begin
    p.x:=novo;
end;
function GetX;
begin
    GetX:=p.x;
end;
procedure SetY;
begin
    p.y:=novo;
end;
function GetY;
begin
    GetY:=p.y;
end;
procedure Prikaz;
begin
    writeln('Horizon. strana prav. iznosi ',p.x);
    writeln('Vertik. strana prav. iznosi ',p.y);
end;
procedure GetPravougaonik;
begin
    pr:=p;
end;
end.
```

### program tp2:

```
uses j2,Crt;
var prav:Pravougaonik;
begin
    clrscr;
    SetX(3); SetY(6);
    Prikaz;
    GetPravougaonik(prav);
    prav.x:=11;
    Prikaz;
    writeln('prav.x=',prav.x);
    readln
end.
```

Kada u programu napišemo SetX(3) i SetY(6) ne vidi se na koju promenljivu se to odnosi. Unit (modul) u Turbo Pascalu može predstavljati tip podatka, ali taj tip nije modul. Da bi modul bio tip i da bi tip bio modul potrebno je nadgraditi pojam sloga i uvesti pojam objekta (*object*). Objekat je slog u kome se mogu naći procedure i funkcije koje operišu nad elementima sloga.

```
TYPE Pravougaonik=object
        x,y:integer;
    end;
```

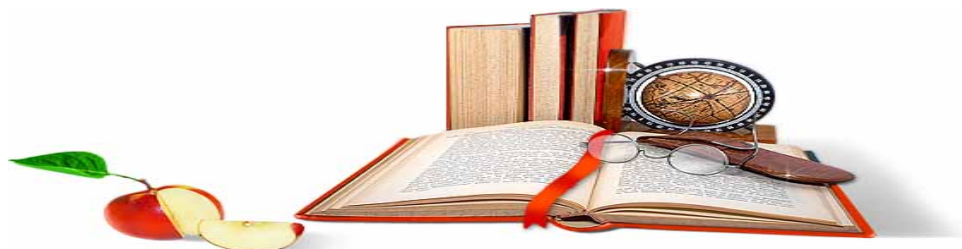
Ovakav objekat je novi tip podatka i primerke tog tipa definišemo naredbom VAR:

```
VAR p:Pravougaonik;
```

---- OSTATAK TEKSTA NIJE PRIKAZAN. CEO RAD MOŽETE PREUZETI NA SAJTU [WWW.MATURSKI.NET](http://WWW.MATURSKI.NET) ----

[BESPLATNI GOTOVI SEMINARSKI, DIPLOMSKI I MATURSKI TEKST](http://WWW.SEMINARSKIRAD.ORG)  
RAZMENA LINKOVA - RAZMENA RADOVA  
RADOVI IZ SVIH OBLASTI, POWERPOINT PREZENTACIJE I DRUGI EDUKATIVNI MATERIJALI.

[WWW.SEMINARSKIRAD.ORG](http://WWW.SEMINARSKIRAD.ORG)  
[WWW.MAGISTARSKI.COM](http://WWW.MAGISTARSKI.COM)  
[WWW.MATURSKIRADOVI.NET](http://WWW.MATURSKIRADOVI.NET)



NA NAŠIM SAJTOVIMA MOŽETE PRONAĆI SVE, BILO DA JE TO [SEMINARSKI](#), [DIPLOMSKI](#) ILI [MATURSKI](#) RAD, POWERPOINT PREZENTACIJA I DRUGI EDUKATIVNI MATERIJAL. ZA RAZLIKU OD OSTALIH MI VAM PRUŽAMO DA POGLEDATE SVAKI RAD, NJEGOV SADRŽAJ I PRVE TRI STRANE TAKO DA MOŽETE TAČNO DA ODABERETE ONO ŠTO VAM U POTPUNOSTI ODGOVARA. U BAZI SE NALAZE [GOTOVI SEMINARSKI, DIPLOMSKI I MATURSKI RADOVI](#) KOJE MOŽETE SKINUTI I UZ NJIHOVU POMOĆ NAPRAVITI JEDINSTVEN I UNIKATAN RAD. AKO U [BAZI](#) NE NAĐETE RAD KOJI VAM JE POTREBAN, U SVAKOM MOMENTU MOŽETE NARUČITI DA VAM SE IZRADI NOVI, UNIKATAN SEMINARSKI ILI NEKI DRUGI RAD RAD NA LINKU [IZRADA RADOVA](#). PITANJA I ODGOVORE MOŽETE DOBITI NA NAŠEM [FORUMU](#) ILI NA

[maturskiradovi.net@gmail.com](mailto:maturskiradovi.net@gmail.com)