

مجموعه ها:

برای تعریف مجموعه ها از ساختار زیر استفاده می شود

Type

```
<SetName >= set of <EnumType >;
```

با این تعریف نوع مجموعه ای به نام <SetName > که اعضای آن همان داده های غیر عددی نوع

Enum هستند که با <EnumType > مشخص شده مثلاً

Type

```
FontStyle = (fsBold ,fsItalic ,fsUnderlined );
```

```
FontStyles = set of FontStyle;
```

فرامین آماده ای برای کار با مجموعه ها در دلفی وجود دارد که می توان برای مجموعه تعریف شده از

آنها استفاده کرد. مثلاً دستوری برای اجتماع وجود دارد.

```
Include(< Name >,< EnumMember >);
```

این دستور عضوی را به

مجموعه < Name > اضافه می کند یا برای تفاضل

```
Exclude(< Name >,< EnumMember >);
```

و مشابهاً و معادل دو دستور بالا

```
< Name >:=< Name >+ [< EnumMember >];
```

در دستور بالا گروه بکار رفته منظور انتخابی بودن عنصر درون آن نیست بلکه خود گروه باید

گذاشته شود مثلاً

```

var
  a : FontStyle;
  b : FontStyle;
M
begin
  D
  b := b + a; // = Include(b, a)

```

یا

```

D
b := b - a; // = Exclude(b, a)

```

اگر می خواستیم همین مجموعه را توسط روش عدد n بیتی شبیه سازی کنیم باید دستورات زیر را

استفاده می کردیم:

```

Const
  fsBold : Cardinal = $00000001;
  fsItalic : Cardinal = $00000002;
  fsUnderlined : Cardinal = $00000004;

```

```

var
  a : Cardinal;
  b : Cardinal;
begin

  a := fsBold;
  b := b or a; // Equivalent to [fsBold];
  یا
  b := b and (not a); // Equivalent to Include.
  // Equivalent to exclude
end.

```

لذا این تمهیدات بوضوح در دو جهت قرار داده شده اند. یکی نزدیک تر شدن زبان برنامه نویسی به

زبان گویش انسانی دیگری اختصاص حداقل حافظه لازم به متغیرها. در مثال فوق برای سه متغیر از نوع عدد

حسابی *cardinal* که 64 بیتی است استفاده کردیم. سه داده بیش از 3 بیت نیاز ندارد و از طرفی فقط می

توان مضارب صحیح 8 بیت حافظه را اختصاص داد (مضارب صحیح بایت). این مدیریت حافظه در هنگام

استفاده از *Enum* ها و مجموعه ها بوضوح راحت تر است.

شبکه رشد - شبکه ملی مدارس ایران



Olympiad.roshd.ir