

8: Indexers

12/01/2003

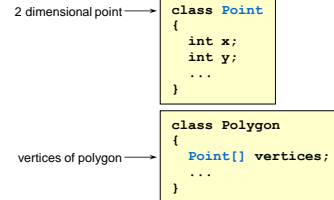
Indexers

(modified)



Polygon

- Polygon might contain array of vertices
 - vertex represented as Point class object



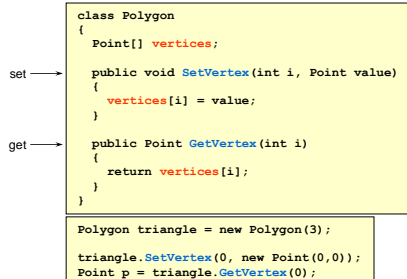
developmentor



2

Vertex access

- Polygon might provide access methods for vertices



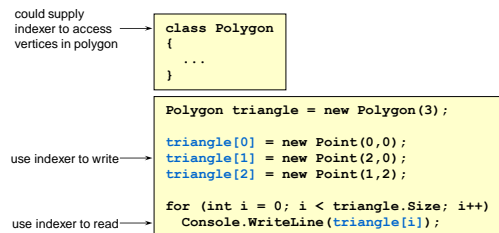
developmentor



3

Indexer

- Indexer allows a type to support array-like indexing



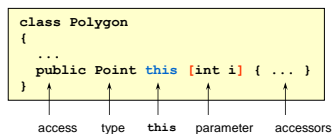
developmentor



4

Indexer definition syntax

- Indexer definition specifies:
 - access level
 - type of data being indexed
 - keyword **this**
 - parameters in []
 - accessors in { }



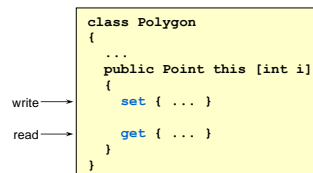
developmentor



5

Accessor options

- Can control access by selectively providing accessors
 - **set** for write access
 - **get** for read access
 - both for read/write
 - must provide at least one



developmentor



6

8: Indexers

12/01/2003

Implementing set accessor

- To implement set accessor:
 - index in call passed as parameter
 - data passed in hidden argument called **value**

```
class Polygon
{
    Point[] vertices;

    public Point this [int i]
    {
        set
        {
            vertices[i] = value;
        }
        ...
    }
    ...
}
```

index →

data →

calls set accessor → `triangle[2] = new Point(1, 2);`

developmentor



Implementing get accessor

- To implement get accessor:
 - index in call passed as parameter
 - use **return** to send out data of specified type

```
class Polygon
{
    ...
    public Point this [int i]
    {
        get
        {
            return vertices[i];
        }
        ...
    }
    ...
}
```

return a Point →

calls get accessor → `Point apex = triangle[2];`

developmentor



Indexer parameters

- Indexer parameters
 - can be any type
 - must be passed by value, cannot be passed **ref** or **out**

```
class Department
{
    public Employee this [string name] ...
    ...
}
```

string parameter →

pass string →

```
Department d = new Department();
...
Employee e = d["Ann"];
```

developmentor



Indexer parameter number

- Indexers can have one or more parameters

```
class Matrix
{
    public int this [int row, int column] ...
    ...
}
```

2 parameters →

pass 2 indices →

```
Matrix m = new Matrix();
...
int v = m[1,2];
```

developmentor



Indexer overloading

- Indexers can be overloaded
 - must differ in number or type of parameters

```
class Matrix
{
    public int[] this [int row] ...
    public int this [int row, int column] ...
    ...
}
```

entire row →

single element →

developmentor



No static indexers

- Indexers cannot be static
 - must be applied to object

```
class Polygon
{
    public static Point this [int i] { ... }
    ...
}
```

error →

developmentor



8: Indexers

12/01/2003

Indexer implementation

- Indexers are compiled into property named `Item`
 - C# programs must use indexer
 - exposed to other .NET languages as property

compiled to property
named `Item` →

```
class Polygon
{
    public Point this [int i]
    {
        ...
    }
}
```

developmentor



13