Class School

This class defines a `School` object.

Implements IEnumerable to allow AcademicYear objects inside the School object to be iterated through.

Author: Yashwant Rathor

Inheritance

- System.Object
  - School

Implements

System.Collections.IEnumerable

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: SchoolReportSystem.model

Assembly: SchoolReportSystem.dll

Syntax

```
public class School : IEnumerable
```

Constructors

School(String, Int32)

This custom constructor is responsible for creating a School object.

Declaration

```
public School(string name, int year)
```

Parameters

Type	Name	Description
System.String	*name*	The school name.
System.Int32	*year*	The year number.

Examples

```
School sl = new School("Portsborough Secondary", 8);
```

This creates a School object with the name as 'Portsborough Secondary' and year as '8'.

School(String, Int32, Int32)

This alternate custom constructor is responsible for creating a School object.

Declaration

```
public School(string name, int first, int last)
```

Parameters

Type	Name	Description
System.String	*name*	The school name.
System.Int32	*first*	The lowest school year.
System.Int32	*last*	The highest school year

Examples

```
School sl = new School("Earlmount High", 7, 11);
```

This creates a School object with name as 'Earlmount High', with years 7 to 11.

Properties

GetSchoolName

This method retrieves the school's name.

Declaration

```
public string GetSchoolName { get; }
```

Property Value

Type	Description
System.String	returns the value of GetSchoolName.

Examples

```
School sl = new School("Earlmount High", 7, 11);
sl.GetSchoolName;
```

This will return "Earlmount High".

GetYears

This method retrieves the school years.

Declaration

```
public List<AcademicYear> GetYears { get; }
```

Property Value

Type	Description
System.Collections.Generic.List<AcademicYear>	returns the value of GetYears.

Examples

```
School sl = new School("Earlmount High", 7, 11);
sl.GetYears;
```

This will return {(7), (8), (9), (10), (11), (12), (13)}.

Methods

AddYear(AcademicYear)

This method stores a valid AcademicYear object into the School object.

Declaration

```
public void AddYear(AcademicYear y)
```

Parameters

Type	Name	Description
AcademicYear	*y*	AcademicYear object to be added to the school.

Examples

```
School sl = new School("Nottingham Secondary", 13);
sl.AddYear(12);
```

Year 12 has been added to 'sl'.

AddYears(Int32, Int32)

This method stores a range of valid AcademicYear objects into the School object, dependent on the 'min' and 'max' parameters.

Declaration

```
public void AddYears(int min, int max)
```

Parameters

Type	Name	Description
System.Int32	*min*	The minimum inclusive number value for the range.
System.Int32	*max*	The maximum inclusive number value for the range.

Examples

```
School sl = new School("Berkshire High", 7, 8);
sl.AddYears(9, 13);
```

Years: 9, 10, 11, 12 and 13 have been added to 'sl'.

GetEnumerator()

This method returns the IEnumerator for the School class.

Declaration

```
public IEnumerator GetEnumerator()
```

Returns

Type	Description
System.Collections.IEnumerator	returns the iteration of the non-generic collection for a School object.

GetYearByNo(Int32)

This method retrieves the AcademicYear object which is equal to 'yearNo'.

Declaration

```
public AcademicYear GetYearByNo(int yearNo)
```

Parameters

Type	Name	Description
System.Int32	*yearNo*	

Returns

Type	Description
AcademicYear	returns the year with the matching year number specifed by the 'yearNo' parameter.

Examples

```
School sl = new School("Midwest College", 9, 11);
sl.AddYears(12, 13);
```

sl.GetYearByNo(13); Year 13 will be returned.

ToString()

This method overrides the default 'ToString()' representation of the School class.

Declaration

```
public override string ToString()
```

Returns

Type	Description
System.String	

Overrides

System.Object.ToString()

Examples

```
School sl = new School("Holloway High", 7, 13);
sl.ToString();
```

Implements

System.Collections.IEnumerable