

Namespace DocumentingTheStudent AccountingSystem

Classes

[Student](#)

Класс, представляющий информацию о студенте.

[StudentDatabase](#)

Класс для работы с базой данных студентов.

Enums

[EducationType](#)

Тип обучения студента.

Enum EducationType

Namespace: [DocumentingTheStudentAccountingSystem](#)

Assembly: laba8.dll

Тип обучения студента.

```
public enum EducationType
```

Fields

Budget = 0

Обучение на бюджетной основе.

Contract = 1

Обучение на контрактной (платной) основе.

Class Student

Namespace: [DocumentingTheStudentAccountingSystem](#)

Assembly: laba8.dll

Класс, представляющий информацию о студенте.

```
public class Student
```

Inheritance

[object](#)  ← Student

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Constructors

Student(string, string, int, string, EducationType)

Конструктор для создания нового студента.

```
public Student(string firstName, string lastName, int age, string studentId,  
EducationType education)
```

Parameters

firstName [string](#) 


Имя студента.

lastName [string](#) 

Фамилия студента.

age [int](#) 

Возраст студента.

studentId [string](#) 

Номер студенческого билета.

`education` [EducationType](#)

Тип обучения студента.

Properties

Age

Возраст студента.

```
public int Age { get; set; }
```

Property Value

[int](#)

Education

Статус студента.

```
public EducationType Education { get; set; }
```

Property Value

[EducationType](#)

FirstName

Имя студента.

```
public string FirstName { get; set; }
```

Property Value

[string](#)

LastName

Фамилия студента.

```
public string LastName { get; set; }
```

Property Value

[string](#) 

StudentID

Номер студенческого билета.

```
public string StudentID { get; set; }
```

Property Value

[string](#) 

Methods

GetEducationTypeInRussian()

Возвращает тип обучения на русском языке.

```
public string GetEducationTypeInRussian()
```

Returns

[string](#) 

Тип обучения на русском языке.

Class StudentDatabase


Namespace: [DocumentingTheStudentAccountingSystem](#)

Assembly: laba8.dll








Класс для работы с базой данных студентов.

```
public class StudentDatabase
```

Inheritance

[object](#)  ← StudentDatabase

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#) 

Methods

AddStudent(Student)

Метод для добавления студента в базу.

```
public void AddStudent(Student student)
```

Parameters

student [Student](#)

Студент для добавления.

Examples

```
var db = new StudentDatabase();  
var student1 = new Student("Иван", "Петров", 20, "S12345", EducationType.Budget);  
db.AddStudent(student1); // В данном примере добавляется новый студент Иван Петров.
```

Exceptions

[ArgumentNullException](#)

Выбрасывается, если студент равен null.

FindStudent(string)

Метод для поиска студента по номеру студенческого билета.

```
public Student FindStudent(string studentId)
```

Parameters

studentId [string](#)

Номер студенческого билета студента для поиска.

Returns

[Student](#)

Найденный студент или null, если студент не найден.

Examples

```
var db = new StudentDatabase();  
var student1 = new Student("Иван", "Петров", 20, "S12345");  
var foundStudent = db.FindStudent("S12345"); // Найдет студента Ивана
```

GetAllStudents()

Получение списка всех студентов.

```
public List<Student> GetAllStudents()
```

Returns

[List](#) <[Student](#)>

Список студентов.

Examples

```
var allStudents = db.GetAllStudents(); // Получит список всех студентов
```

RemoveStudent(string)

Метод для удаления студента по номеру студенческого билета.

```
public void RemoveStudent(string studentId)
```

Parameters

studentId [string](#)

Номер студенческого билета студента для удаления.

Examples

```
var db = new StudentDatabase();  
var student1 = new Student("Иван", "Петров", 20, "S12345");  
db.RemoveStudent("S12345"); // Удалит Ивана из базы
```

Exceptions

[InvalidOperationException](#)

Выбрасывается, если студент с указанным номером не найден.

SortStudentsByLastName()

Метод для сортировки студентов по фамилии. Алгоритм сортировки использует метод OrderBy из LINQ для упрощения сортировки списка студентов по их фамилии.

```
public void SortStudentsByLastName()
```

Examples

```
db.SortStudentsByLastName(); // Сортирует студентов по фамилии
```