**Училище**

**Общ преглед**

Във вашата фирма постъпва проект за създаване на приложение, обслужващо училище.

Вашият софтуер трябва да описва **училище** **(**School**)** и **ученик (**Student**).**

Tрябва да реализирате функционалност, която да позволява някои основни команди с учениците в училището. Поредицата от команди приключва с „**END**”. За ваше удобство ще получите готов **Program.cs** файл, и ще трябва да реализирате само необходимите класове **School.cs** и **Student.cs**

**Подзадача 1: 30 точки**

**Student**

Всички ученици имат име и списък с оценки:

name – низ, съставен от малки и/или големи латински букви

grade – оценка (дробно число)

|  |
| --- |
| Student.cs |
| private string name;  private List<double> grades;    public Student()  {    }    public Student(string name, double grade)  {      //TODO: Добавете вашия код тук …  }  public string Name  {     //TODO: Добавете вашия код тук …  }  public double Grade  {    //TODO: Добавете вашия код тук …  }    public override string ToString()  {    //TODO: Добавете вашия код тук …  } |

**School**

Всички училища имат име и списък с ученици:

name – низ, съставен от малки и/или големи латински букви

students – списък от ученици

|  |
| --- |
| School.cs |
| private string name;  private List<Student> students;    public School()  {    }    public School (string name)  {     //TODO: Добавете вашия код тук …  }    public string Name  {     //TODO: Добавете вашия код тук …  }    public List<Student> Students  {     //TODO: Добавете вашия код тук …  }    public void AddStudent (string name, double grade)  {     //TODO: Добавете вашия код тук …  }    public double AverageResultInRange(int start, int end)  {     //TODO: Добавете вашия код тук …  }      public List<string> RemoveStudentsByGrade(double grade)  {     //TODO: Добавете вашия код тук …  }    public List<Student> SortAscendingByName()  {     //TODO: Добавете вашия код тук …  }    public List<Student> SortDescendingByGrade()  {     //TODO: Добавете вашия код тук …  }    public bool CheckStudentIsInSchool(string name)  {     //TODO: Добавете вашия код тук …  }    public string[] ProvideInformationAboutAllStudents()  {     //TODO: Добавете вашия код тук …  } |

**Командa за добавяне на ученик**

Вашето приложение трябва да обслужва следната команда за добавяне на ученик:

* **Add** <**име оценка**> - тази команда има за цел да добави ученик с неговото име и оценка.

**Команда за извеждане на информация**

Вашето приложение във всеки един момент може да получи заявка да отпечата информация за всички ученици. Командата за това е следната:

* **Print** - отпечатва информация за всички ученици в структурата във формат:  
  **Student {name} has {grade}.**
* За успешна реализация трябва да реализирате ваша версия на **ToString()**метода за класа **Student.**
* **Оценката трябва да бъде форматирана до втория знак след десетичната запетая.**
* **SortByName** - Трябва да сортира учениците в списъка по име във възходящ (нарастващ) ред. След което трябва да бъде върнат сортираният списък.
* **SortByGrade** - Трябва да сортира учениците в списъка по оценка в низходящ (намаляващ) ред.  След което трябва да бъде върнат сортираният списък.
* **CheckStudent<name**> - Трябва да бъде намерен ученик, на който името отговаря на подаденото. При успешно намиране  трябва да върнете **true,**а в обратен случай **false.**

**Подзадача 2: 30 точки**

**Подзадача 3: 20 точки**

* **RemoveStudents <оценка**> - Трябва да бъдат намерени всички ученици от списъка, на които оценката им е по-ниска от дадената в командата. Да се върне списък с имената на имената на намерените ученици.
* **AverageResult<int startIndex, int endIndex**> - Трябва да бъдат взети оценките на  всички ученици от **startIndex** до **endIndex** и да се намери средноаритметичната им стойност.
* Програмата ще получава множество редове с информация. Всеки ред представлява команда. Самият вход се обработва изцяло от примерния Program.cs.
* Всички команди приключват с въвеждането на END

**Подзадача 4: 20 точки**

**Вход**

**Изход**

За някои от командите не е нужно да извеждате нищо. За други е необходимо форматиране на изход – напр. Product.ToString(), Product.Name()

**Ограничения**

* Всички реални числа ще бъдат в диапазона **–10000.** до **+10000.**
* Имената няма да съдържат интервал

**Примери**

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| Add Peter 4.50  Add Gergana 2.30  Add Georgi 3.40  Print  END | Added student Peter.  Added student Gergana.  Added student Georgi.  Student Peter has 4.50.  Student Gergana has 2.30.  Student Georgi has 3.40. |
| Add Archie 4.50  Add Reggie 2.30  Add Betty 3.40  Add Jug 3.50  Add Clarke 4.90  Add Veronica 3.40  Print  SortByName  Print  SortByGrade  Print  CheckStudent Betty  CheckStudent Jughead  AverageResult 0 4  RemoveStudents 3.60  END | Added student Archie.  Added student Reggie.  Added student Betty.  Added student Jug.  Added student Clarke.  Added student Veronica.  Student Archie has 4.50.  Student Reggie has 2.30.  Student Betty has 3.40.  Student Jug has 3.50.  Student Clarke has 4.90.  Student Veronica has 3.40.  First student is: Archie  Student Archie has 4.50.  Student Betty has 3.40.  Student Clarke has 4.90.  Student Jug has 3.50.  Student Reggie has 2.30.  Student Veronica has 3.40.  The worst student is: Reggie  Student Clarke has 4.90.  Student Archie has 4.50.  Student Jug has 3.50.  Student Betty has 3.40.  Student Veronica has 3.40.  Student Reggie has 2.30.  Student Betty is available.  Student Jughead is not available.  Average result: 3.94  Poor students: Jug, Betty, Veronica, Reggie |

**Точки**

Разбивката по подзадачи е следната:

1. **30** точки
2. **30** точки
3. **20** точки
4. **20** точки

Общ брой точки: **100**