**School**

**Avaliable from**: Sunday, 5 July 2020, 10:00  
**Due date**: Sunday, 5 July 2020, 16:00  
**Requested files**: Program.cs, School.cs, Student.cs ([Download](https://it-kariera.mon.bg/e-learning/mod/vpl/views/downloadrequiredfiles.php?id=4881))  
**Type of work**: Individual work  
**Настройки на оценките**: Максимална оценка: 100  
**Run**: Не. **Evaluate**: Да. **Evaluate just on submission**: Да  
**Automatic grade**: Да.

**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**

**Requested files**

**Program.cs**

using System;

using System.Collections.Generic;

namespace Exam1

{

class Program

{

static School school = new School("School1");

static void Main(string[] args)

{

string line;

while ("END" != (line = Console.ReadLine()))

{

string[] cmdArgs = line.Split(' ');

switch (cmdArgs[0])

{

case "Add":

AddStudent(cmdArgs[1], double.Parse(cmdArgs[2]));

break;

case "AverageResult":

AverageResultInRange(int.Parse(cmdArgs[1]), int.Parse(cmdArgs[2]));

break;

case "RemoveStudents":

RemoveStudentsByGrade(double.Parse(cmdArgs[1]));

break;

case "SortByName":

SortAscendingByName();

break;

case "SortByGrade":

SortDescendingByGrade();

break;

case "CheckStudent":

CheckStudentIsInSchool(cmdArgs[1]);

break;

case "Print":

ProvideInformationAboutAllStudents();

break;

}

}

}

private static void ProvideInformationAboutAllStudents()

{

string[] info = school.ProvideInformationAboutAllStudents();

foreach (string item in info)

{

Console.WriteLine(item);

}

}

private static void CheckStudentIsInSchool(string name)

{

if (school.CheckStudentIsInSchool(name))

{

Console.WriteLine($"Student {name} is available.");

}

else

{

Console.WriteLine($"Student {name} is not available.");

}

}

private static void SortDescendingByGrade()

{

school.SortDescendingByGrade();

Console.WriteLine("The worst student is: " + school.Students[school.Students.Count - 1].Name);

}

private static void SortAscendingByName()

{

school.SortAscendingByName();

Console.WriteLine("First student is: " + school.Students[0].Name);

}

private static void RemoveStudentsByGrade(double grade)

{

List<string> leftStudents = school.RemoveStudentsByGrade(grade);

Console.WriteLine("Poor students: " + string.Join(", ", leftStudents));

}

private static void AverageResultInRange(int start, int end)

{

double averageGrade = school.AverageResultInRange(start, end);

Console.WriteLine($"Average result: {averageGrade:f2}");

}

private static void AddStudent(string name, double grade)

{

school.AddStudent(name, grade);

Console.WriteLine($"Added student {name}.");

}

}

}

#### School.cs

#### Student.cs

## Execution files

#### vpl\_run.sh

#!/bin/bash

# This file is part of VPL for Moodle - http://vpl.dis.ulpgc.es/

# Script for running C# language

# Copyright (C) 2014 Juan Carlos Rodríguez-del-Pino

# License http://www.gnu.org/copyleft/gpl.html GNU GPL v3 or later

# Author Juan Carlos Rodríguez-del-Pino <jcrodriguez@dis.ulpgc.es>

# @vpl\_script\_description Using mcs

# load common script and check programs

. common\_script.sh

check\_program mcs

check\_program mono

if [ "$1" == "version" ] ; then

echo "#!/bin/bash" > vpl\_execution

echo "mcs --version" >> vpl\_execution

chmod +x vpl\_execution

exit

fi

get\_source\_files cs

# compile

export MONO\_ENV\_OPTIONS=--gc=sgen

MONO\_REFERENCES=-r:System.Numerics.dll

mcs -pkg:dotnet $MONO\_REFERENCES -out:output.exe $SOURCE\_FILES

if [ -f output.exe ] ; then

cat common\_script.sh > vpl\_execution

echo "export MONO\_ENV\_OPTIONS=--gc=sgen" >> vpl\_execution

echo "mono output.exe \$@" >> vpl\_execution

chmod +x vpl\_execution

grep -E "System\.Windows\.Forms" output.exe &>/dev/null

if [ "$?" -eq "0" ] ; then

mv vpl\_execution vpl\_wexecution

fi

fi

#### vpl\_evaluate.cases

Case=Case000-001

input=Add Peter 4.50

Add Gergana 2.30

Add Georgi 3.40

Print

END

output=Added student Peter.

Added student Gergana.

Added student Georgi.

Student Peter has 4.50.

Student Gergana has 2.30.

Student Georgi has 3.40.

Case=Case000-002

input=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

output=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

Case=Case001

Fail message= Case1\_fail

grade reduction=10

input=Add Peter 4.50

Add Gergana 2.30

Add Georgi 3.40

Add Todor 4.90

Add Pesho 3.40

Add Ivan 4.50

Print

END

output=Added student Peter.

Added student Gergana.

Added student Georgi.

Added student Todor.

Added student Pesho.

Added student Ivan.

Student Peter has 4.50.

Student Gergana has 2.30.

Student Georgi has 3.40.

Student Todor has 4.90.

Student Pesho has 3.40.

Student Ivan has 4.50.

Case=Case002

Fail message= Case2\_fail

grade reduction=10

input=Add Peter 4.50

Add Gergana 2.30

Print

Add Georgi 3.40

Print

Add Todor 4.90

Add Pesho 3.40

Add Ivan 4.50

Print

END

output=Added student Peter.

Added student Gergana.

Student Peter has 4.50.

Student Gergana has 2.30.

Added student Georgi.

Student Peter has 4.50.

Student Gergana has 2.30.

Student Georgi has 3.40.

Added student Todor.

Added student Pesho.

Added student Ivan.

Student Peter has 4.50.

Student Gergana has 2.30.

Student Georgi has 3.40.

Student Todor has 4.90.

Student Pesho has 3.40.

Student Ivan has 4.50.

Case=Case003

Fail message= Case3\_fail

grade reduction=10

input=Add Peter 4.50

Print

Add Gergana 2.30

Print

Add Georgi 3.40

Print

Add Todor 4.90

Print

Add Pesho 3.40

Print

Add Ivan 4.50

Print

END

output=Added student Peter.

Student Peter has 4.50.

Added student Gergana.

Student Peter has 4.50.

Student Gergana has 2.30.

Added student Georgi.

Student Peter has 4.50.

Student Gergana has 2.30.

Student Georgi has 3.40.

Added student Todor.

Student Peter has 4.50.

Student Gergana has 2.30.

Student Georgi has 3.40.

Student Todor has 4.90.

Added student Pesho.

Student Peter has 4.50.

Student Gergana has 2.30.

Student Georgi has 3.40.

Student Todor has 4.90.

Student Pesho has 3.40.

Added student Ivan.

Student Peter has 4.50.

Student Gergana has 2.30.

Student Georgi has 3.40.

Student Todor has 4.90.

Student Pesho has 3.40.

Student Ivan has 4.50.

Case=Case004

Fail message= Case4\_fail

grade reduction=10

input=Add Peter 4.50

Add Gergana 2.30

Add Georgi 3.40

Add Todor 4.90

Add Pesho 3.40

Add Ivan 4.50

SortByName

Print

END

output=Added student Peter.

Added student Gergana.

Added student Georgi.

Added student Todor.

Added student Pesho.

Added student Ivan.

First student is: Georgi

Student Georgi has 3.40.

Student Gergana has 2.30.

Student Ivan has 4.50.

Student Pesho has 3.40.

Student Peter has 4.50.

Student Todor has 4.90.

Case=Case005

Fail message= Case5\_fail

grade reduction=10

input=Add Peter 4.50

Add Gergana 2.30

Add Georgi 3.40

Add Todor 4.90

Add Pesho 3.40

Add Ivan 4.50

SortByGrade

Print

END

output=Added student Peter.

Added student Gergana.

Added student Georgi.

Added student Todor.

Added student Pesho.

Added student Ivan.

The worst student is: Gergana

Student Todor has 4.90.

Student Peter has 4.50.

Student Ivan has 4.50.

Student Georgi has 3.40.

Student Pesho has 3.40.

Student Gergana has 2.30.

Case=Case006

Fail message= Case6\_fail

grade reduction=10

input=Add Peter 4.50

Add Gergana 2.30

Add Georgi 3.40

Add Todor 4.90

Add Pesho 3.40

Add Ivan 4.50

CheckStudent Gergana

CheckStudent Todor

CheckStudent Ivanka

CheckStudent Pesho

CheckStudent Stefan

CheckStudent Misho

Add Misho 3.60

CheckStudent Misho

Add Ivanka 6.00

CheckStudent Ivanka

END

output=Added student Peter.

Added student Gergana.

Added student Georgi.

Added student Todor.

Added student Pesho.

Added student Ivan.

Student Gergana is available.

Student Todor is available.

Student Ivanka is not available.

Student Pesho is available.

Student Stefan is not available.

Student Misho is not available.

Added student Misho.

Student Misho is available.

Added student Ivanka.

Student Ivanka is available.

Case=Case007

Fail message= Case7\_fail

grade reduction=10

input=Add Peter 4.50

Add Gergana 2.30

Add Georgi 3.40

Add Todor 4.90

Add Pesho 3.40

Add Ivan 4.50

Print

RemoveStudents 4.00

END

output=Added student Peter.

Added student Gergana.

Added student Georgi.

Added student Todor.

Added student Pesho.

Added student Ivan.

Student Peter has 4.50.

Student Gergana has 2.30.

Student Georgi has 3.40.

Student Todor has 4.90.

Student Pesho has 3.40.

Student Ivan has 4.50.

Poor students: Gergana, Georgi, Pesho

Case=Case008

Fail message= Case8\_fail

grade reduction=10

input=Add Peter 4.50

Add Gergana 6.00

Add Georgi 3.40

Add Todor 4.90

Add Pesho 3.40

Add Ivan 4.50

Print

RemoveStudents 6.00

END

output=Added student Peter.

Added student Gergana.

Added student Georgi.

Added student Todor.

Added student Pesho.

Added student Ivan.

Student Peter has 4.50.

Student Gergana has 6.00.

Student Georgi has 3.40.

Student Todor has 4.90.

Student Pesho has 3.40.

Student Ivan has 4.50.

Poor students: Peter, Georgi, Todor, Pesho, Ivan

Case=Case009

Fail message= Case9\_fail

grade reduction=10

input=Add Peter 4.50

Add Gergana 2.30

Add Georgi 3.40

Add Todor 4.90

Add Pesho 3.40

Add Ivan 4.50

Add Reni 3.80

Add Simon 2.90

AverageResult 2 5

END

output=Added student Peter.

Added student Gergana.

Added student Georgi.

Added student Todor.

Added student Pesho.

Added student Ivan.

Added student Reni.

Added student Simon.

Average result: 4.05

Case=Case010

Fail message= Case10\_fail

grade reduction=10

input=Add Peter 4.50

Add Gergana 2.30

Add Georgi 3.40

Add Todor 4.90

Add Pesho 3.40

Add Ivan 4.50

Add Reni 3.80

Add Simon 2.90

Print

AverageResult 0 7

END

output=Added student Peter.

Added student Gergana.

Added student Georgi.

Added student Todor.

Added student Pesho.

Added student Ivan.

Added student Reni.

Added student Simon.

Student Peter has 4.50.

Student Gergana has 2.30.

Student Georgi has 3.40.

Student Todor has 4.90.

Student Pesho has 3.40.

Student Ivan has 4.50.

Student Reni has 3.80.

Student Simon has 2.90.

Average result: 3.71