Рябов Алексей КИТ-119а

**Лабораторная работа №4**

***Тема:*** Свойства класса. Обработка строк. StringBuilder

**Индивидуальное задание:**

* Для автоматического расчёта значений при чтении использовать свойства "только для чтения" (без реализации метода set).
* Вывод для выбранного студента названия группы (аббревиатура названия факультета, номер специальности, год поступления, индекс).
* Вывод для выбранного студента номера курса и семестра на текущий момент.
* Расчёт и вывод для выбранного студента возраста на текущий момент (до дня).
* Продемонстрировать эффективное использование StringBuilder для обработки строк.

**Ход работы:**

* Добавлены свойства для «Студента», что рассчитываются автоматически, в том числе с использованием StringBuilder:

namespace Lab4

{

[Serializable]

public class DataStudLab4 : Lab1.DataStud

{

public string IndexGroup => this.Faculty + "-" + this.Speciality + "-" + this.ReciptDate.Year + "-" + Group;

public int Semester

{

get

{

var y = DateTime.Now.Year - ReciptDate.Year;

var m = DateTime.Now.Month;

if (y == 0)

{

if (m < 8)

{

return 0;

}

}

if (y > 4)

{

return 8;

}

if (m > 8)

{

return y \* 2 + 1;

}

return y \* 2;

}

}

public int Course => Semester / 2 + Semester % 2;

public string Old

{

get

{

var time = DateTime.Now.Ticks - this.Birthday.Ticks;

var old = new DateTime(time - 315360000000000);

return old.ToString();

}

}

public string Faculty

{

set;

get;

}

public DataStudLab4() : base()

{

Faculty = "KIT";

}

public DataStudLab4(string name, string surName, string patronimic, DateTime birthday,

DateTime reciptDate, string group, int speciality, float academicPerformance, string faculty)

: base(name, surName, patronimic, birthday, reciptDate, group, speciality, academicPerformance)

{

Faculty = faculty;

}

public DataStudLab4(string name, string surName, string patronimic)

: base(name, surName, patronimic)

{

Faculty = "KIT";

}

public override bool Equals(object obj)

{

if (obj == null)

{

return false;

}

if (!(obj is DataStudLab4 temp))

{

return false;

}

if (Name.CompareTo(temp.Name) != 0)

{

return false;

}

if (Surname.CompareTo(temp.Surname) != 0)

{

return false;

}

if (Patronimic.CompareTo(temp.Patronimic) != 0)

{

return false;

}

if (Faculty.CompareTo(temp.Faculty) != 0)

{

return false;

}

if (!Birthday.Equals(temp.Birthday))

{

return false;

}

if (!ReciptDate.Equals(temp.ReciptDate))

{

return false;

}

if (Speciality != temp.Speciality)

{

return false;

}

if (AcademicPreformance != temp.AcademicPreformance)

{

return false;

}

return true;

}

public override int GetHashCode()

{

return base.GetHashCode();

}

}

}

Рисунок 4.1 – Код программы

* Запустил программу и получил результат:



Рисунок 4.2 – Вывод.

***Выводы:*** разработал программу на С#. Изучил методы обработки строк.