Министерство образования Республики Беларусь

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники

Кафедра «Системы автоматизированного проектирования»

Отчёт по лабораторной работе

по дисциплине «Визуальные средства разработки программных приложений»

то теме «Потоки»

|  |  |
| --- | --- |
| Выполнил: | Студенты группы 10702315  Ломтев  Жевняк |
| Проверил: | Красновская С.В. |

##### Минск

##### БНТУ

##### 2018

**Цель:**

* Познакомиться с классами потоков (streams) и вспомогательными классами для чтения из потоков и записи в них.
* Реализовать класс или классы потоков или вспомогательные классы в соответствии с заданием.

**Задание:**

***4****. BeepingWriter и BeepingReader*

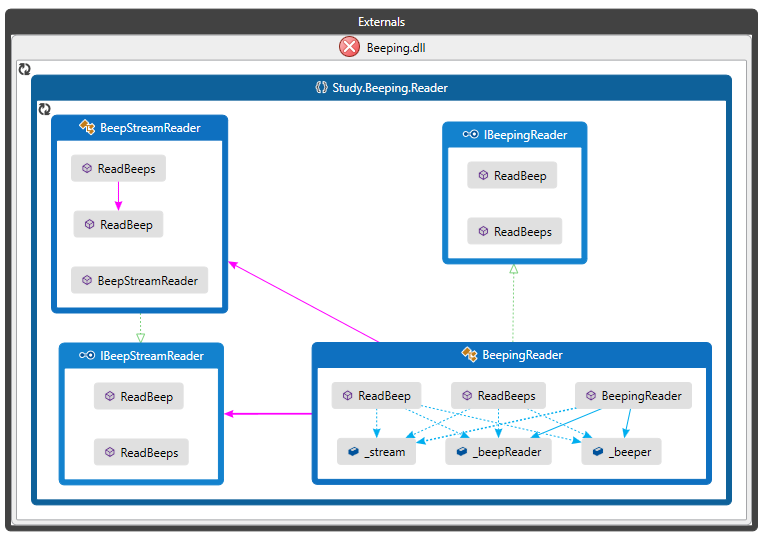
а) BeepingWriter должен иметь метод WriteBeep(Beep object), записывающий в файловый поток переданный объект и производящий с помощью Console.Beep(...) соответствующий звуковой сигнал.

б) BeepingReader должен иметь метод Beep ReadBeep(), считывающий из файла (файлового потока) объект типа Beep и при этом производящий с помощью Console.Beep(...) соответствующий звуковой сигнал. Также нужен метод, считывающий файл до конца. Возвращаемое значение этого метода - список объектов типа Beep.

Beep - должен содержать продолжительность (duration) сигнала и его частоту (frequency).

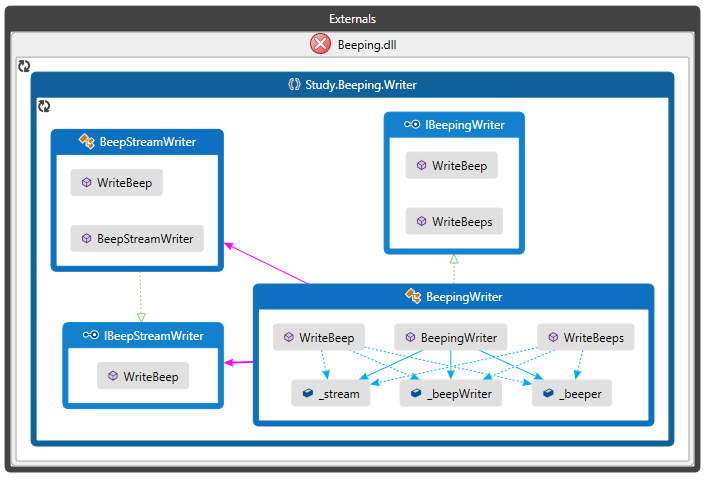
**Результат выполнения лабораторной работы:**

В ходе выполнения лабораторной работы были реализованные классы BeepingWriter и BeepingReader. Оба класса используют абстрактный класс потоков Stream, что расширяет функционал, а также не противоречит условию задачи, т.к. класс FileStream (поток, работающий с файлами) является одной из реализаций класса Stream.



Player

Writer



Reader

**Листинг:**

public class BeepingReader

: IBeepingReader

{

private IBeeper \_beeper;

private IBeepStreamReader \_beepReader;

private Stream \_stream;

public BeepingReader(

Stream stream,

IBeeper beeper,

IBeepStreamReader beepReader

) {

\_beeper = beeper;

\_beepReader = beepReader;

\_stream = stream;

\_stream.Position = 0;

}

public Beep ReadBeep()

{

Beep beep = \_beepReader.ReadBeep(\_stream);

\_beeper.Beep(beep);

return beep;

}

public IEnumerable<Beep> ReadBeeps()

{

List<Beep> beeps = new List<Beep>();

Beep beep = null;

do

{

beep = \_beepReader.ReadBeep(\_stream);

if (beep != null)

{

beeps.Add(beep);

\_beeper.Beep(beep);

}

}

while (beep != null);

return beeps;

}

}

public class BeepingWriter

: IBeepingWriter

{

private IBeeper \_beeper;

private IBeepStreamWriter \_beepWriter;

private Stream \_stream;

public BeepingWriter(

Stream stream,

IBeeper beeper,

IBeepStreamWriter beepWriter

) {

\_stream = stream;

\_beeper = beeper;

\_beepWriter = beepWriter;

}

public void WriteBeep(Beep @object)

{

\_beepWriter.WriteBeep(

\_stream,

@object

);

\_beeper.Beep(@object);

}

public void WriteBeeps(IEnumerable<Beep> beeps)

{

foreach (Beep beep in beeps)

{

\_beepWriter.WriteBeep(

\_stream,

beep

);

\_beeper.Beep(beep);

}

}

}