1. **Введение**

Объектом разработки является программное средство “Генератор гармонических рядов” – приложение, предназначенное для наглядного изображения гармонических рядов, упрощения понимания этой темы, а также, в некотором роде, исследований.

Среда разработки: Visual Studio 2017 Commuity.

Цель: изучение Объектно-ориентированного программирования.

1. **Аналитический обзор литературы по теме проекта. Изучение требований, определение вариантов использования.**

По причине того, что в основе приложения будет лежать язык C# и WPF, литература, которая будет представлена ниже максимально должна раскрывать суть этих основ.

* *Windows 8 Apps Revealed Using XAML and C#: Using XAML and C# - Adam Freeman*
* *Лекции по математическому анализу – Кудрявцев Н. Л.*

Правила, которыми должно придерживаться приложение

* Удобство пользования интерфейсом программы;
* Простой и интуитивно понятный доступ к информации;
* Возможность каждому пользователю работать с программой;
* Корректная работа программы и достоверность информации.

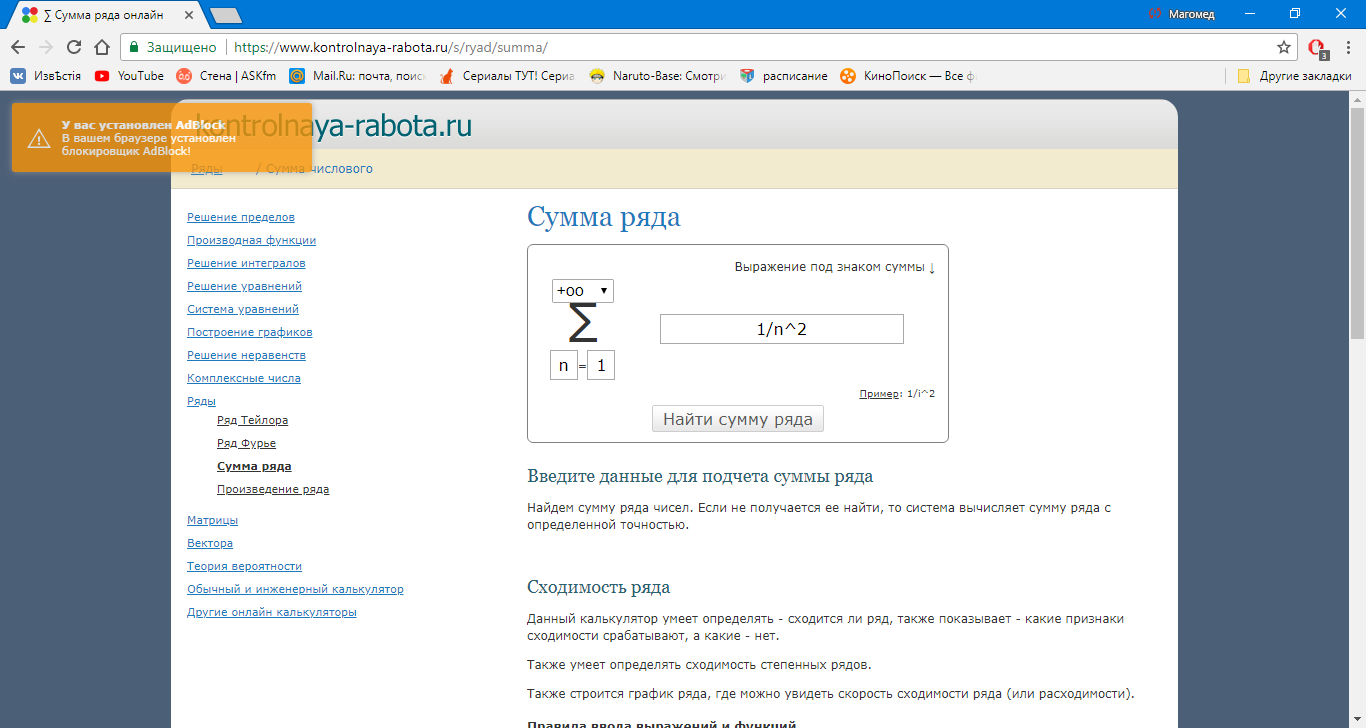
Данное приложение будет использоваться преимущественно студентами. Думаю, что это полезная программа, которая максимально упрощает процесс обучения и некоторых исследований.

# **Постановка задачи**

Нашей задачей является генерация ряда, что подразумевает работу со случайными величинами. Задачей пользователя будет ввод желаемого количества данных, программа должна будет сделать оставшуюся работу. Алгоритм прост: обработка введенных пользователем данных и генерирование недостающих, после чего – вывод результата в специальное окно. Нужные вычисления (частичная, полная сумма и т.д.) выводятся в соответствующие области.

1. **Обзор прототипов**

[**https://www.kontrolnaya-rabota.ru/s/ryad/summa/**](https://www.kontrolnaya-rabota.ru/s/ryad/summa/)



Из всех найденных прототипов были найдены лишь аналоги вот такой программы: нахождение суммы ряда по введенным данным. В таких программах велик процент синтаксических ошибок, т.к. неподготовленному пользователю трудно ввести данные сложного ряда корректно. Я учту это и минимизирую количество критических мест в программе.

Задача, как я считаю весьма актуальна, т.к. все имеющиеся программы-конкуренты имеют весьма скудный функционал и не имеют возможности удовлетворить всех потребностей пользователя.

# **Разработка архитектуры проекта**

Имеются 3 группы пользователей:

* Новичок (отсутствие знаний по теме "ряды")
* Продвинутый (базовые знания)
* Эксперт (знание множества тонкостей)

У каждой из этих групп существуют свои права доступа к программе. Новичку после входа в программу будет предложено ознакомиться с теорией (сделано это, естественно, для того, чтобы пользователь знал, что он делает). Ему будет доступен базовый функционал. Продвинутому будет предложена теория немного иного уровня, нежели новичку, как и функционал. Эксперту будет доступен полный функционал, исчерпывающая информация о теме и полный контроль над генерацией ряда. Каждый пользователь может переходить на любой уровень по своему желанию, но для этого нужно набрать достаточно баллов в специальном тесте (для перехода на уровень ниже или регистрации "новичка" тест не требуется).

Подробнее о функциях:

1. Новичок:

* Сгенерировать абсолютно случайный ряд
* Сгенерировать ряд с указанием степени многочленов и коэффициентов
* Вычислить n-ый член ряда

2. Продвинутый:

* Функционал новичка
* Сгенерировать знакопеременный ряд
* Вычислить частичную сумму первых n элементов
* Управление сходимостью
* Вычислить полную сумму ряда
* Генерация степенного и функционального рядов и вычисление области сходимости

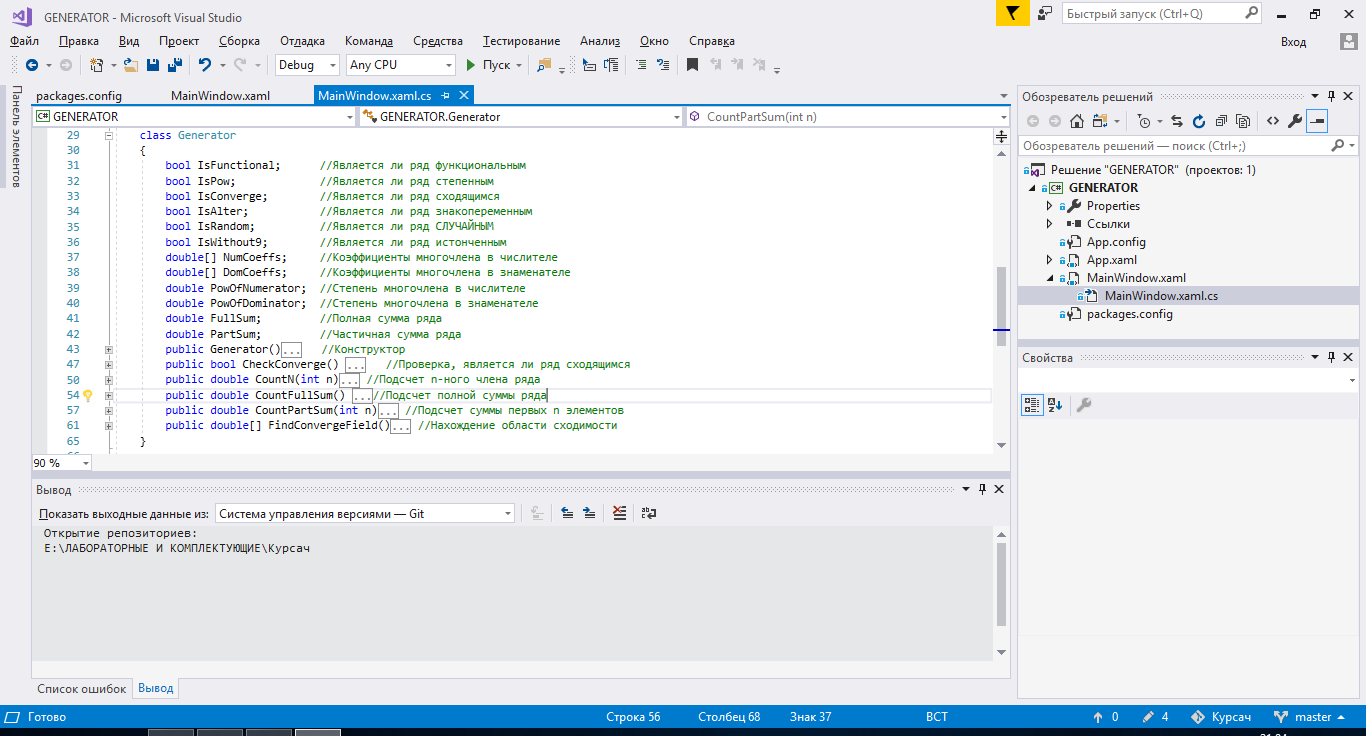
3. Эксперт:

* Функционал продвинутого
* Построение графика
* Сгенерировать случайный гармонический ряд
* Сгенерировать истонченный гармонический ряд

4. Все пользователи:

* Просмотреть историю генерации
* Пройти тест и перейти на следующий уровень
* Просмотреть теорию

Структура класса Generator (класс, содержащий всю информацию и ряде):



*Рис. 1 – структура класса “Generator”*