# Notes about the MAL Deviation Program

Table of Contents

[1 Notes about the MAL Deviation Program 1](#_Toc525461924)

[3 Concept 1](#_Toc525461925)

[4 User interface 1](#_Toc525461926)

[5 Open tasks 1](#_Toc525461927)

[6 Task list 1](#_Toc525461928)

[7 Problems and open questions 1](#_Toc525461929)

[8 Class descriptions 1](#_Toc525461930)

[8.1 Form 1 1](#_Toc525461931)

[8.1.1 Fields 1](#_Toc525461932)

[8.1.2 Properties 1](#_Toc525461933)

[8.1.3 Constructors 1](#_Toc525461934)

[8.1.4 Private Methods 1](#_Toc525461935)

[8.1.5 Events 1](#_Toc525461936)

[8.2 Anime 1](#_Toc525461937)

[8.2.1 Fields 1](#_Toc525461938)

[8.2.2 Properties 1](#_Toc525461939)

[8.2.3 Constructors 1](#_Toc525461940)

[8.3 Stats 1](#_Toc525461941)

[8.3.1 Fields 1](#_Toc525461942)

[8.3.2 Properties 1](#_Toc525461943)

[8.3.3 Constructors 1](#_Toc525461944)

[8.3.4 Private Methods 1](#_Toc525461945)

[8.3.5 Public Methods 1](#_Toc525461946)

[8.4 Scores 1](#_Toc525461947)

[8.4.1 Fields 1](#_Toc525461948)

[8.4.2 Properties 1](#_Toc525461949)

[8.4.3 Constructors 1](#_Toc525461950)

# Concept

This program is meant to download and analyze the statistics of anime series from the website myanimelist.net. Main part of the analysis is the calculation of the deviation of the votes for each series. After downloading and analysis is done, the application is to report interesting values of most interest, for example the biggest and smallest deviation and their corresponding anime series.

Since series are indexed by the website through a unique number, the user can set a start and stop index, in which between the program will search for anime series.

# User interface

The interface consists of the following elements:

* A dialog to select a save folder, in which the end results will be saved in
* Selectors for setting the start and stop index
* Start, pause and stop buttons
* Text output for reporting current and end results
* Progress bar
* Label for indicating the current status of the program

# Open tasks

* During scanning, the user interface should stay responsive. Therefore, the scanning must be done in a background task. But the background task must be able to report back to the main task and the main task must then update the interface.
* Progress should be saved during the scanning not just at the end. The program should be able to detect already scanned series and skip them.

# Task list

* Understand current state of the project
* Add scanning functionality
  + Inform myself about Background worker
    - Inform myself about events

# Problems and open questions

# Class descriptions

## Form 1

### Fields

|  |  |  |
| --- | --- | --- |
| String | \_folderPath | Stores the path to te folder, in which the statistics will be saved |
| Int32 | \_currentIndex | Anime series and films are accessed by a number in the URL. This field stores that number. |
| List<Anime> | \_animeList | A List of all found anime within the given range |
| Status | \_currentStatus | Status of the program |

### Properties

|  |  |  |
| --- | --- | --- |
| Status | CurrentStatus | Gets and sets the internal field \_currentStatus |

### Constructors

|  |
| --- |
| Form1() |

#### Form1()

This constructor initializes the main application window.

### Private Methods

|  |  |  |
| --- | --- | --- |
| void | buttonSavePath\_Click(object sender, EventArgs e) |  |
| void | buttonTest\_Click(object sender, EventArgs e) |  |
| void | buttonStart\_Click(object sender, EventArgs e) |  |
| void | buttonStop\_Click(object sender, EventArgs e) |  |
| void | run() |  |
| void | stop() |  |
| void | pause() |  |

#### buttonSavePath\_Click(object sender, EventArgs e)

Description

This method gets called when the button labled “Browse” is clicked. It opens a dialog for the user to select a folder.  
The path to the selected folder will be shown in the label.

Parameters

|  |  |  |
| --- | --- | --- |
| object | sender |  |
| EventArgs | e |  |

Returns

**Void**

#### buttonTest\_Click(object sender, EventArgs e)

Description

This method gets called when the test button is clicked. Its content can be used for debugging.

Parameters

|  |  |  |
| --- | --- | --- |
| object | sender |  |
| EventArgs | e |  |

Returns

**Void**

#### buttonStart\_Click(object sender, EventArgs e)

Description

This method gets called when the start button is clicked. It starts or pauses the search and data retrieve process.

Parameters

|  |  |  |
| --- | --- | --- |
| object | sender |  |
| EventArgs | e |  |

Returns

**Void**

#### buttonStop\_Click(object sender, EventArgs e)

Description

This method gets called when the stop button is clicked. It stops the search.

Parameters

|  |  |  |
| --- | --- | --- |
| object | sender |  |
| EventArgs | e |  |

Returns

**Void**

#### run()

Description

This method sets the status label to “Running” and the start button text to “Pause”. It will also set the progressbar value.

Parameters

**None**

Returns

**Void**

#### stop()

Description

This method sets the status label to “Stopped” and the start button text to “Start”.

Parameters

**None**

Returns

**Void**

#### pause()

Description

This method sets the status label to “Paused” and the start button text to “Start”.

Parameters

**None**

Returns

**Void**

### Events

## Anime

### Fields

|  |  |  |
| --- | --- | --- |
| Int32 | \_id | Individual index of the anime |
| String | \_name | Name of the anime |
| Stats | \_stats | Statistics of the anime |

### Properties

|  |  |  |
| --- | --- | --- |
| Int32 | Id | Gets and sets the internal field \_id |
| String | Name | Gets and sets the internal field \_name |
| Stats | Stats | Gets and sets the internal field \_stats |

### Constructors

|  |
| --- |
| Anime(Int32 id, String name, Stats stats) |

#### Anime(Int32 id, String name, Stats stats)

This constructor initializes the fields \_id, \_name and \_stats to the corresponding parameters.

## Stats

### Fields

|  |  |  |
| --- | --- | --- |
| Score[] | \_scores | Number of votes for each rating |
| Double | \_mean | Mean of all votings |
| Double | \_deviation | Deviation of the votes |
| Int32 | \_totalVotes | Total number of votes |

### Properties

|  |  |  |
| --- | --- | --- |
| Score[] | Scores | Gets and sets the internal field \_scores |
| Double | Mean | Gets the internal field \_mean |
| Double | Deviation | Gets the internal field \_deviation |
| Int32 | TotalVotes | Gets the internal field \_totslVotes |

### Constructors

|  |
| --- |
| Stats() |
| Stats(Score[] scores) |

#### Stats()

This constructor iterates through the \_scores[] field and initializes it to default values.

#### Stats(Score[] scores)

This constructor initializes the field \_scores[] with the values given in scores.

### Private Methods

|  |  |  |
| --- | --- | --- |
| Void | getMean() | Calculates the means of all votes |
| Void | getDeviation() | Calculates the deviation of all votes |
| Void | getTotalVotes() | Sums up all votes |

#### getMean()

Description

This method calculates the mean value by iterating over all scores and saving the result in the internal field \_mean.

Parameters

**None**

Returns

**Void**

#### getDeviation()

Description

This method calculates the deviation. A more detailed description of the maths involved will follow in a later chapter. The result is stored in the internal field \_deviation.

Parameters

**None**

Returns

**Void**

#### getTotalVotes()

Description

This method somes the votes from ich rating (1 to 10) and stores the result in the internal field \_totalVotes.

Parameters

**None**

Returns

**Void**

### Public Methods

|  |  |  |
| --- | --- | --- |
| Void | update() | Updates the fields |

#### update()

Description

This method calls the private methods getMean(), getDeviation() and getTotalVotes() to update the corresponding internal fields.

Parameters

**None**

Returns

**Void**

## Scores

### Fields

|  |  |  |
| --- | --- | --- |
| Double | \_percent | Percentage of votes from total number of votes |
| Int32 | \_votes | Numbe of votes for the score |

### Properties

|  |  |  |
| --- | --- | --- |
| Double | Percent | Gets and sets the internal field \_percent |
| Int32 | Votes | Gets and sets the internal field \_votes |

### Constructors

|  |
| --- |
| Score() |
| Score(Double percent, Int32 votes) |

#### Score()

This constructor initializes the fields \_percent and \_votes to zero.

#### Score(Double percent, Int32 votes)

This constructor initializes the fields \_percent and \_votes to the values of percent and votes.