**OsuDbApi**

Оглавление

[Общие типы данных 1](#_Toc60423267)

[Enum GameplayMode 1](#_Toc60423268)

[Osu!.Db 1](#_Toc60423269)

[Сlass OsuDbReader 1](#_Toc60423270)

[Сlass Beatmap 2](#_Toc60423271)

[Struct IntDoublePair 3](#_Toc60423272)

[Struct TimingPoint 3](#_Toc60423273)

[Enum RankedStatus 3](#_Toc60423274)

[Enum UserPermissions 4](#_Toc60423275)

[Collection.db 4](#_Toc60423276)

[Class CollectionDbReader 4](#_Toc60423277)

[Class CollectionDbWritter 4](#_Toc60423278)

[Class BeatmapCollection 5](#_Toc60423279)

[Scores.db 5](#_Toc60423280)

[Сlass ScoresDbReader 5](#_Toc60423281)

[Class BeatmapScores 5](#_Toc60423282)

[Сlass Score 6](#_Toc60423283)

[Enum Mods 6](#_Toc60423284)

Общие типы данных

Enum GameplayMode

**namespace** OsuDbApi.Enums

**Значения:**

* Standart или STD = 1;
* Taiko = 2;
* CTB = 4;
* Mania = 8.

Osu!.Db

Сlass OsuDbReader

**namespace OsuDbApi.OsuDb**

Реализован интерфейс IDisposable

**Конструкторы:**

* OsuDbReader(string osuDbFile).

**Методы:**

|  |  |  |
| --- | --- | --- |
| **Возвращаемый тип** | **Метод** | **Назначение** |
| Bool | Next(); | Читает следующую запись Beatmap из файла в буфер и возвращает значение, указывающее успешность операции. |
| Beatmap | GetValue() | Возвращает прочитанное значение из буфера |
| - | Dispose() | Освобождает все ресурсы, связанные с текущим объектом |

**Свойства:**

|  |  |  |
| --- | --- | --- |
| **Тип** | **Свойство** | **Описание** |
| **get;** | | |
| Bool | IsDisposed | Указывает, были ли освобождены все ресурсы, связанные с текущим объектом |
| Int | BeatmapReadCount | Количество прочитанных карт |
| OsuVersion | Версия osu! |
| FolderCount | Количество папок |
| Bool | AccountUnlocked | Если false, учетная запись заблокирована |
| DateTime | DateAccountUnlocked | Дата разблокировки учетной записи |
| String | PlayerName | Имя игрока |
| Int | BeatmapsCount | Количество битмапов |
| UserPermissions | UserPermissions | Разрешения пользователя |
| String | OsuDbFile | Имя файла, включая путь к нему, с которым работает текущий экземпляр OsuDbReader |

Сlass Beatmap

**namespace OsuDbApi.OsuDb.Models**

**Конструкторы:**

* Beatmap();

**Свойства:**

|  |  |  |
| --- | --- | --- |
| **Тип** | **Свойство** | **Описание** |
| **get; set;** | | |
| Int? | SizeInBytes | Размер записи beatmap в байтах. Присутствует только в том случае, если версия меньше 20191106. |
| String | ArtistName | Имя исполнителя |
| ArtistNameUnicode | Имя исполнителя, в Юникоде |
| SongTitle | Название песни |
| SongTitleUnicode | Название песни, в Юникоде |
| CreatorName | Имя создателя |
| Difficulty | Сложность |
| AudioFileName | Имя аудиофайла |
| Md5Hash | MD5 хэш битмапа |
| OsuFileName | Имя файла osu, соответствующего этой битмапе |
| RankedStatus | RankedStatus | Ранк статус |
| Short | HitcirclesCount | Количество нот |
| SlidersCount | Количество слайдеров |
| SpinnersCount | Количество спинеров |
| DateTime | LastModificationTime | Время последнего изменения карты |
| Float | ApproachRate | AR карты |
| CircleSize | CS карты |
| HpDrain | HP drain |
| OverallDifficulty | OD карты |
| Double | SliderVelocity | Скорость слайдеров |
| List<IntDoublePair> | StarRatingStandart | Информация о star-рейтинге для osu! Standard. Null, если версия меньше 20140609. |
| StarRatingTaiko | Информация о star-рейтинге для osu! Taiko. Null, если версия меньше 20140609. |
| StarRatingCtb | Информация о star-рейтинге для osu! CTB. Null, если версия меньше 20140609. |
| StarRatingMania | Информация о star-рейтинге для osu! Mania. Null, если версия меньше 20140609. |
| TimeSpan | DrainTime | Время слива |
| TotalTime | Общее время |
| AudioPreviewTime | Время, когда начинается предварительный просмотр аудио |
| List<TimingPoint> | TimingPoints | Временные точки |
| Int | Id | Beatmap ID |
| SetId | Beatmap set ID |
| ThreadId | Thread ID |
| Byte | GradeAchievedStandard | Ранг, достигнутый в osu! Standard |
| GradeAchievedTaiko | Ранг, достигнутый в osu! Taiko |
| GradeAchievedCtb | Ранг, достигнутый в osu! CTB |
| GradeAchievedMania | Ранг, достигнутый в osu! Mania |
| Short | LocalOffset | Локальный offset битовой карты |
| Float | StackLeniency | Стек снисходительности |
| GameplayMode | GameplayMode | Игровой режим |
| String | SongSource | Источник песни |
| SongTags | Теги песен |
| Short | OnlineOffset | Онлайн-offset |
| String | FontTitleSong | Шрифт, используемый для названия песни |
| Bool | IsUnplayed | Если true, то данная карта ещё не сыграна юзером |
| DateTime | LastTimePlay | В последний раз, когда битмап игралась |
| Bool | IsOsz2 | Является ли битмап osz2 |
| String | FolderName | Имя папки beatmap относительно папки Songs |
| DateTime | LastTimeCheckedRepository | Дата последней проверки карты с загруженной на репозиторий оригиналом |
| Bool | IgnoreSound | Если true, игнорировать битмап звук |
| IgnoreSkin | Если true, игнорировать скин beatmap |
| DisableStoryboard | Если true, отключить storyboard |
| DisableVideo | Если true, отключить видео |
| VisualOverride | Если true, визуальное переопределение |
| Byte | ManiaScrollSpeed | Скорость прокрутки Mania |

Struct IntDoublePair

**namespace** OsuDbApi.OsuDb.Models;

**Конструкторы:**

* IntDoublePair();
* IntDoublePair(int intValue, double doubleValue).

**Свойства:**

|  |  |  |
| --- | --- | --- |
| **Тип** | **Свойство** | **Описание** |
| **get;** | | |
| int | IntValue | Целое число |
| double | DoubleValue | Число с плавающей запятой |

Struct TimingPoint

**namespace** OsuDbApi.OsuDb.Models

**Конструкторы:**

* TimingPoint(double bpm, double offset, bool isInherit).

**Свойства:**

|  |  |  |
| --- | --- | --- |
| **Тип** | **Свойство** | **Описание** |
| **get;** | | |
| double | BPM | BPM |
| Offset | Смещение в песне (в миллисекундах) |
| bool | IsInherit | Указывает, наследуется эта временная точка или нет, если true то наследуется |

Enum RankedStatus

**namespace OsuDbApi.OsuDb.Enums**

**Значения:**

* Unknown = 0;
* Unsubmitted = 1;
* PendingWipGraveyard = 2;
* Unused = 3;
* Ranked = 4;
* Approved = 5;
* Qualified = 6;
* Loved = 7.

Enum UserPermissions

**namespace OsuDbApi.OsuDb.Enums**

**Значения:**

* None = 0;
* Normal = 1;
* Moderator = 2;
* Supporter = 4;
* Friend = 8;
* Peppy = 16;
* WorldCupStaff = 32.

Collection.db

Class CollectionDbReader

**namespace OsuDbApi.CollectionDb**

Реализован интерфейс IDisposable

**Конструкторы:**

* CollectionDbReader(string collectionDbFile).

**Методы:**

|  |  |  |
| --- | --- | --- |
| **Возвращаемый тип** | **Метод** | **Назначение** |
| Bool | Next() | Читает следующую запись BeatmapCollection из файла в буфер и возвращает значение, указывающее успешность операции. |
| BeatmapCollection | GetValue() | Возвращает прочитанное значение из буфера |
| - | Dispose() | Освобождает все ресурсы, связанные с текущим объектом |

**Свойства:**

|  |  |  |
| --- | --- | --- |
| **Тип** | **Свойство** | **Описание** |
| **get;** | | |
| Bool | IsDisposed | Указывает, были ли освобождены все ресурсы, связанные с текущим объектом |
| Int | BeatmapCollectionReadCount | Количество прочитанных коллекций |
| OsuVersion | Версия osu! |
| BeatmapCollectionsCount | Количество коллекций |
| String | CollectionDbFile | Абсолютный путь к файлу collection.db, с которым работает текущий экземпляр CollectionDbReader |

Class CollectionDbWritter

**namespace OsuDbApi.CollectionDb**

**Конструкторы:**

* CollectionDbWritter(string collectionDbFile);
* CollectionDbWritter(string collectionDbFile, int osuVersion).

Если аргумент osuVersion отсутствует, будет установлено дефолтное значение: 20201210.

**Методы:**

|  |  |  |
| --- | --- | --- |
| **Возвращаемый тип** | **Метод** | **Назначение** |
| - | Save() | Записывает все данные в файл collection.db |

**Свойства:**

|  |  |  |
| --- | --- | --- |
| **Тип** | **Свойство** | **Описание** |
| **get;** | | |
| List<BeatmapCollection> | BeatmapCollections | Список коллекций для записи |
| Int | OsuVersion | Версия osu! |
| String | CollectionDbFile | Абсолютный путь к файлу collection.db, с которым работает текущий экземпляр CollectionDbWritter |

Class BeatmapCollection

**namespace OsuDbApi.CollectionDb.Models**

**Конструкторы:**

* BeatmapCollection().

**Свойства:**

|  |  |  |
| --- | --- | --- |
| **Тип** | **Свойство** | **Описание** |
| **get; set;** | | |
| String | Name | Название коллекции |
| List<string> | Beatmaps | Список хешей карт входящих в данную коллекцию |

Scores.db

Сlass ScoresDbReader

**namespace OsuDbApi.ScoresDb**

Реализован интерфейс IDisposable

**Конструкторы:**

* ScoresDbReader(string scoresDbFile).

**Методы:**

|  |  |  |
| --- | --- | --- |
| **Возвращаемый тип** | **Метод** | **Назначение** |
| Bool | Next() | Читает следующую запись BeatmapScores из файла в буфер и возвращает значение, указывающее успешность операции. |
| BeatmapScores | GetValue() | Возвращает прочитанное значение из буфера |
| - | Dispose() | Освобождает все ресурсы, связанные с текущим объектом |

**Свойства:**

|  |  |  |
| --- | --- | --- |
| **Тип** | **Свойство** | **Описание** |
| **get;** | | |
| Bool | IsDisposed | Указывает, были ли освобождены все ресурсы, связанные с текущим объектом |
| Int | BeatmapScoresReadCount | Количество прочитанных Beatmap скоров |
| OsuVersion | Версия osu! |
| BeatmapScoresCount | Количество Beatmap скоров |
| String | ScoresDbFile | Абсолютный путь к файлу scores.db, с которым работает текущий экземпляр ScoresDbReader |

Class BeatmapScores

**namespace OsuDbApi.ScoresDb.Models**

**Конструкторы:**

* BeatmapScores().

**Свойства:**

|  |  |  |
| --- | --- | --- |
| **Тип** | **Свойство** | **Описание** |
| **get; set;** | | |
| String | BeatmapHash | Beatmap хеш |
| List<Score> | Scores | Скоры на этой beatmap |

Сlass Score

**namespace OsuDbApi.ScoresDb.Models**

**Конструкторы:**

* Score().

**Свойства:**

|  |  |  |
| --- | --- | --- |
| **Тип** | **Свойство** | **Описание** |
| **get; set;** | | |
| GameplayMode | GameplayMode | Игровой режим |
| Int | ScoreVersion | Версия скора (повтора) |
| String | BeatmapHash | Хеш beatmap |
| PlayerName | Имя игрока |
| Md5Hash | Хеш записи |
| Short | Count300 | Количество 300-х |
| Count100 | Количество 100-х в Osu!Standard, 150-х в Taiko, 100-х в CTB, 100-х в Mania |
| Count50 | Количество 50-х в Osu!Standard, маленький фрукт в CTB, 50-х в Mania |
| GekiCount | Количество Geki в Osu!Standard, Макс 300-x в Mania |
| KatuCount | Количество Katu в Osu!Standard, 200-х в Mania |
| MissCount | Количество промахов |
| Int | ReplayScore | Скор |
| Short | MaxCombo | Максимальное Комбо |
| Bool | PerfectCombo | Если true, идеальное карта сыграна на FC (Full Combo) |
| Int | CombinationModsUsed | Побитовая комбинация модов |
| DateTime | TimestampReplay | Дата, когда скор был поставлен |
| Long | OnlineScoreId | Онлайн-идентификатор скора |

Enum Mods

**namespace OsuDbApi.ScoresDb.Enums**

**Значения:**

* None = 0;
* NoFail = 1;
* Easy = 2;
* TouchDevice = 4;
* Hidden = 8;
* HardRock = 16;
* SuddenDeath = 32;
* DoubleTime = 64;
* Relax = 128;
* HalfTime = 256;
* Nightcore = 512;
* Flashlight = 1024;
* Autoplay = 2048;
* SpinOut = 4096;
* Autopilot = 8192;
* Perfect = 16384;
* Key1 = 67108864;
* Key2 = 268435456;
* Key3 = 134217728;
* Key4 = 32768;
* Key5 = 65536;
* Key6 = 131072;
* Key7 = 262144;
* Key8 = 524288;
* Key9 = 16777216;
* KeyMod = 1015808;
* FadeIn = 1048576;
* Random = 2097152;
* Cinema = 4194304;
* TargetPractice = 8388608;
* Coop = 33554432.