Класс filesystem

Редакция от 28.04.2026 · docs.kb-agava.ru

Содержание

1 Методы 2

Класс предоставляет интерфейс для взаимодействия с файловой системой операционной системы. Позволяет выполнять операции с файлами и директориями, получать информацию о них, а также управлять текущей рабочей директорией.

Класс доступен начиная с версии `AS 1.6+`.

1 Методы

```
bool changeCurrentPath(const string &in path)
```

Назначение: Изменяет текущую рабочую директорию.

Аргументы:

- `path` – путь к новой директории (относительный или абсолютный).

Возвращаемое значение:

- `true` – если путь существует и текущая директория успешно изменена.
- `false` – если указанный путь не существует или не является директорией.

Примечания:

- Если путь задан относительно, он объединяется с текущей директорией.
- Путь не должен содержать обратные слеши `\` – они автоматически заменяются на `/`.
- Завершающие слеши автоматически удаляются из пути.
- Не поддерживает разрешение внутренних ссылок `/./` и `/../`.

```
string getCurrentPath() const
```

Назначение: Возвращает текущую рабочую директорию.

Аргументы: Отсутствуют.

Возвращаемое значение:

- Строка с абсолютным путём текущей директории, использующая разделители `/`.

Примечания:

- При создании объекта `filesystem` текущая директория инициализируется рабочим каталогом приложения.
- Все обратные слеши `\` заменяются на `/` для единообразия.

```
array<string> @getDirs() const
```

Назначение: Получает список всех поддиректорий внутри текущей директории.

Аргументы: Отсутствуют.

Возвращаемое значение:

- Массив строк с именами поддиректорий.

Примечания:

- Специальные записи `.` (текущая директория) и `..` (родительская директория) исключаются из результата.
- Возвращает пустой массив, если директорий нет или произошла ошибка чтения.

```
array<string> @getFiles() const
```

Назначение: Получает список всех файлов внутри текущей директории.

Аргументы: Отсутствуют.

Возвращаемое значение:

- Массив строк с именами файлов.

Примечания:

- В результат включаются только файлы, директории исключаются.
- Возвращает пустой массив, если файлов нет или произошла ошибка чтения.

```
bool isDir(const string &in path) const
```

Назначение: Проверяет, является ли указанный путь директорией.

Аргументы:

- `path` – путь к объекту файловой системы (относительный или абсолютный).

Возвращаемое значение:

- `true` – если объект существует и является директорией.
- `false` – если объект не существует, не является директорией или произошла ошибка.

Примечания:

- Относительные пути разрешаются относительно текущей директории.
- Абсолютные пути определяются наличием двоеточия `:` (например, `C:`) или начального слеша `/` или `\`.

```
bool isLink(const string &in path) const
```

Назначение: Проверяет, является ли указанный путь символической ссылкой (symlink).

Аргументы:

- `path` – путь к объекту файловой системы (относительный или абсолютный).

Возвращаемое значение:

- `true` – если объект существует и является символической ссылкой (на Windows – репарс-точкой).
- `false` – если объект не является ссылкой, не существует или произошла ошибка.

Примечания:

- Кроссплатформенная поддержка: Windows (атрибут `FILE_ATTRIBUTE_REPARSE_POINT`) и POSIX (флаг `S_IFLNK`).

- Относительные пути разрешаются относительно текущей директории.

```
int64 getSize(const string &in path) const
```

Назначение: Возвращает размер файла в байтах.

Аргументы:

- `path` – путь к файлу (относительный или абсолютный).

Возвращаемое значение:

- Размер файла в байтах (тип `int64`).
- `-1` – если файл не существует, указан путь к директории или произошла ошибка.

Примечания:

- На Windows используется API-функция `GetFileSizeEx`.
- На POSIX-системах используется `stat`.
- Для директорий возвращает `-1`.

```
int makeDir(const string &in path)
```

Назначение: Создаёт новую директорию.

Аргументы:

- `path` – путь к создаваемой директории (относительный или абсолютный).

Возвращаемое значение:

- `0` – директория успешно создана.
- `-1` – ошибка создания (например, недостаточно прав, путь не существует).

Примечания:

- **Важно:** Не поддерживает рекурсивное создание промежуточных директорий.
- На Windows создаёт директорию с правами по умолчанию.
- На POSIX создаёт с правами `S_IRWXU | S_IRWXG | S_IROTH | S_IXOTH` (755: чтение/запись/исполнение для владельца, чтение/исполнение для группы и остальных).

```
int removeDir(const string &in path)
```

Назначение: Удаляет пустую директорию.

Аргументы:

- `path` – путь к удаляемой директории (относительный или абсолютный).

Возвращаемое значение:

- `0` – директория успешно удалена.
- `-1` – ошибка удаления (директория не пуста, не существует, недостаточно прав).

Примечания:

- Удаляет только **пустые** директории.
- Не поддерживает рекурсивное удаление с содержимым.

- На Windows используется `RemoveDirectoryW`, на POSIX – `rmdir`.

```
int deleteFile(const string &in path)
```

Назначение: Удаляет файл.

Аргументы:

- `path` – путь к удаляемому файлу (относительный или абсолютный).

Возвращаемое значение:

- `0` – файл успешно удалён.
- `-1` – ошибка удаления (файл не существует, недостаточно прав, указана директория).

Примечания:

- На Windows используется `DeleteFileW`, на POSIX – `unlink`.
- Для удаления директорий используйте `removeDir`.

```
int copyFile(const string &in source, const string &in target)
```

Назначение: Копирует файл из одного местоположения в другое.

Аргументы:

- `source` – путь к исходному файлу.
- `target` – путь к целевому файлу (не директории).

Возвращаемое значение:

- `0` – файл успешно скопирован.
- `-1` – ошибка копирования (исходный файл не существует, недостаточно прав, целевой файл уже существует).

Примечания:

- На Windows используется API-функция `CopyFileW` с параметром `TRUE` (не перезаписывать существующий файл).
- На POSIX-системах копирование выполняется **вручную** побайтово через `fopen/fread/fwrite`, так как стандартной функции копирования нет.
- Целевой файл не должен существовать (флаг `TRUE` на Windows).

```
int move(const string &in source, const string &in target)
```

Назначение: Перемещает или переименовывает файл или директорию.

Аргументы:

- `source` – текущий путь к объекту.
- `target` – новый путь или имя объекта.

Возвращаемое значение:

- `0` – объект успешно перемещён/переименован.
- `-1` – ошибка операции (объект не существует, недостаточно прав).

Примечания:

- Может использоваться как для файлов, так и для директорий.
- На Windows используется `MoveFileW`, на POSIX – `rename`.
- Операция может выполняться в пределах одного тома/файловой системы.

```
datetime getCreateDateTime(const string &in path) const
```

Назначение: Возвращает дату и время создания файла или директории.

Аргументы:

- `path` – путь к объекту файловой системы.

Возвращаемое значение:

- Объект `datetime`, содержащий дату и время создания.

Примечания:

- **Выбрасывает исключение** в скриптовом контексте через `SetException`, если не удаётся получить дату создания.
- На Windows используется `GetFileTime` с полем `createTm`.
- На POSIX-системах используется поле `st_ctime` (change time) – может отличаться от реального времени создания на некоторых файловых системах.

```
datetime getModifyDateTime(const string &in path) const
```

Назначение: Возвращает дату и время последнего изменения файла или директории.

Аргументы:

- `path` – путь к объекту файловой системы.

Возвращаемое значение:

- Объект `datetime`, содержащий дату и время последней модификации.

Примечания:

- Выбрасывает исключение, если не удаётся получить дату изменения.
- На Windows используется `GetFileTime` с полем `modifyTm`.
- На POSIX-системах используется поле `st_mtime` (modify time).