

Класс datetime

Класс доступен начиная с версии AS 1.6+.

□

Содержание

Конструкторы

Конструктор по умолчанию

Конструктор копирования

Конструктор с параметрами

Пример

Методы

Арифметические операции

Операции сравнения

Практические примеры

1 Конструкторы

1.1 Конструктор по умолчанию

```
| datetime dt();
```

Создает объект с текущей датой и временем.

1.2 Конструктор копирования

```
| datetime dt(other);
```

Создает копию существующего объекта datetime.

1.3 Конструктор с параметрами

```
| datetime dt(year, month, day, hour = 0, minute = 0, second = 0);
```

Создает объект с указанной датой и временем.

1.4 Пример

```
| // Только дата
```

```
| datetime newYear = datetime(2024, 1, 1);  
|  
| // Дата и время  
| datetime meeting = datetime(2024, 12, 25, 14, 30, 0);  
|  
| // Полная спецификация  
| datetime exact = datetime(2024, 6, 15, 9, 45, 30);
```

2 Методы

```
| uint get_year() const
```

Возвращает год (4 цифры).

```
| uint get_month() const
```

Возвращает месяц (1-12).

```
| uint get_day() const
```

Возвращает день месяца (1-31).

```
| uint get_hour() const
```

Возвращает часы (0-23).

```
| uint get_minute() const
```

Возвращает минуты (0-59).

```
| uint get_second() const
```

Возвращает секунды (0-59).

Пример:

```
| datetime dt = datetime(2024, 12, 25, 14, 30, 45);  
|  
| uint year = dt.year;    // 2024  
| uint month = dt.month; // 12  
| uint day = dt.day;     // 25  
| uint hour = dt.hour;   // 14  
| uint minute = dt.minute; // 30  
| uint second = dt.second; // 45
```

```
| bool setDate(year, month, day)
```

Устанавливает дату. Возвращает `true` при успехе, `false` при неверной дате.

Пример:

```
| datetime dt;  
| bool success = dt.setDate(2024, 2, 29); // true (високосный год)  
| bool invalid = dt.setDate(2023, 2, 29); // false (не високосный)
```

```
| bool setTime(hour, minute, second)
```

Устанавливает время. Возвращает `true` при успехе, `false` при неверном времени.

Пример:

сpp

```
| datetime dt;  
| bool success = dt.setTime(23, 59, 59); // true  
| bool invalid = dt.setTime(25, 0, 0); // false (часы > 23)
```

3 Арифметические операции

Вычитание дат

```
| int64 difference = dt1 - dt2;
```

Возвращает разницу в секундах.

```
| datetime start = datetime(2024, 1, 1, 0, 0, 0);  
| datetime end = datetime(2024, 1, 2, 0, 0, 0);  
| int64 diff = end - start; // 86400 секунд (1 день)
```

Сложение с секундами

```
| datetime result = dt + seconds;  
| datetime result = seconds + dt; // обратный порядок  
| dt += seconds;
```

Пример:

```
| datetime dt = datetime(2024, 1, 1, 10, 0, 0);  
|  
| // Добавить 2 часа  
| datetime later = dt + 7200;  
| dt += 7200;  
|  
| // Обратный порядок  
| datetime same = 7200 + dt;
```

Вычитание секунд

```
| datetime result = dt - seconds;  
| datetime result = seconds - dt; // обратный порядок  
| dt -= seconds;  
|  
|
```

Пример:

```
| datetime dt = datetime(2024, 1, 1, 10, 0, 0);  
|  
| // Вычесть 30 минут  
| datetime earlier = dt - 1800;  
| dt -= 1800;
```

3.1 Операции сравнения

```
| bool opEquals(other)
```

Проверка равенства дат.

```
| bool opLess(other)
```

Проверка что дата раньше.

```
| int opCmp(other)
```

Сравнение с возвратом -1 (меньше), 0 (равно), 1 (больше).

Пример:

```
| datetime dt1 = datetime(2024, 1, 1);  
| datetime dt2 = datetime(2024, 1, 2);  
|  
| if (dt1 < dt2) {  
|     // dt1 раньше dt2  
| }  
|  
| if (dt1 == dt2) {  
|     // даты равны  
| }  
|  
| int result = dt1.opCmp(dt2); // -1
```

4 Практические примеры

Пример 1: Таймер выполнения

```
| datetime startTime;  
|  
| // Выполняем некоторую операцию  
| for (int i = 0; i < 1000000; i++)  
| {  
|     // какая-то работа  
| }  
|  
| datetime endTime;  
| int64 elapsed = endTime - startTime; // Затраченное время в секундах
```

Пример 2: Расчет дат событий

```
| datetime projectStart = datetime(2024, 1, 15);  
| datetime deadline = datetime(2024, 3, 1);  
|  
| // Вычитание делаем через сложение с отрицательным числом  
| datetime reminder = deadline + (-3 * 86400);  
|  
| // Продление срока на 2 недели (оператор + работает)  
| datetime extendedDeadline = deadline + (14 * 86400);
```

Пример 3: Проверка рабочего времени

```
| datetime currentTime;  
|  
| // Создаем время начала и конца рабочего дня  
| datetime workStart = datetime(currentTime.year, currentTime.month, currentTime.day, 9, 0, 0);  
| datetime workEnd = workStart + 8 * 3600; // 8 часов  
|  
| if (currentTime >= workStart && currentTime <= workEnd)  
| {  
|     //print("Рабочее время");  
| }  
| else  
| {  
|     //print("Вне рабочего времени");  
| }  
| }
```

Пример 4: Валидация даты

```
| bool createAppointment(uint year, uint month, uint day, uint hour)  
| {  
|     datetime appointment;  
|     // Проверяем корректность даты  
|     if (!appointment.setDate(year, month, day))  
|     {  
|         //print("Неверная дата!");  
|         return false;  
|     }  
|     // Проверяем корректность времени  
|     if (!appointment.setTime(hour, 0, 0))  
|     {  
|         //print("Неверное время!");  
|         return false;  
|     }  
|     // Проверяем что дата не в прошлом  
|     datetime now;  
|  
|     if (appointment < now)  
|     {  
|         //print("Дата не может быть в прошлом!");  
|         return false;  
|     }  
|     //print("Встреча создана: " + appointment.year + "-" + appointment.month + "-" + appointment.day);  
|     return true;  
| }  
| }
```

Пример 5: Разбивка интервала времени

```
| datetime start = datetime(2024, 1, 1, 0, 0, 0);  
|
```

```
| datetime end = start + 365 * 86400; // +1 год
|
| int64 totalSeconds = end - start;
| int64 days = totalSeconds / 86400;
| int64 hours = (totalSeconds% 86400) / 3600;
| int64 minutes = (totalSeconds% 3600) / 60;
| int64 seconds = totalSeconds% 60;
|
| //print("Интервал: " + days + " дней, " + hours + " часов, " + minutes + " минут, " + seconds + "
| секунд");
|
```

Источник — https://docs.kb-agava.ru/index.php?title=Класс_datetime&oldid=3527

Эта страница в последний раз была отредактирована 1 июня 2026 в 08:43.