**Знакомимся с классом Date**

[Java Syntax](https://javarush.ru/quests/QUEST_JAVA_SYNTAX)

[Уровень 8](https://javarush.ru/quests/lectures?quest=QUEST_JAVA_SYNTAX&level=8), Лекция 4

[](https://cdn.javarush.ru/images/article/27e7db08-be67-4ce6-9ef0-d7bf672b5ec9/original.jpeg)

— Привет, Амиго. Хочу рассказать тебе про интересный класс – Date. С помощью этого класса можно хранить дату и время, а также измерять временные промежутки.

— Уже интересно. Продолжай.

— Каждый объект типа Date хранит внутри себя время. А время хранится в очень интересном виде – количество миллисекунд, которые прошли с 1 января 1970 года, по Гринвичу.

— Ничего себе!

— Ага. Это число настолько большое, что не влезает в **int**, приходится хранить его в **long**. Зато очень удобно считать разницу между двумя датами: отнял два числа и уже известна разница с точностью до миллисекунд. И к тому же устраняется проблема смены дат и перевода часов.

— А что самое интересное, каждый объект при создании инициализируется текущим временем. Чтобы узнать текущее время – достаточно просто создать объект.

— А как с ним работать?

— Вот тебе несколько примеров:

Получение текущей даты:

public static void main(String[] args) throws Exception

{

**Date** today = new **Date**();

System.out.println("Current date: " + today);

}

Вычисление разницы между двумя датами:

public static void main(String[] args) throws Exception

{

**Date** currentTime = new **Date**(); //получаем текущую дату и время

**Thread**.sleep(3000); //ждём 3 секунды – 3000 миллисекунд

**Date** newTime = new **Date**(); //получаем новое текущее время

long msDelay = newTime.getTime() - currentTime.getTime(); //вычисляем разницу

System.out.println("Time distance is: " + msDelay + " in ms");

}

Наступило ли уже некоторое время:

public static void main(String[] args) throws Exception

{

**Date** startTime = new **Date**();

long endTime = startTime.getTime() + 5000; // +5 секунд

**Date** endDate = new **Date**(endTime);

Thread**.sleep**(3000); // ждем 3 секунды

**Date** currentTime = new **Date**();

**if**(currentTime.after(endDate))//проверяем что время currentTime после endDate

{

System.out.println("End time!");

}

}

Сколько прошло времени с начала сегодняшнего дня:

public static void main(String[] args) throws Exception

{

**Date** currentTime = **new** **Date**();

**int** hours = currentTime.getHours();

**int** mins = currentTime.getMinutes();

**int** secs = currentTime.getSeconds();

System.out.println("Time from midnight " + hours + ":" + mins + ":" + secs);

}

Сколько дней прошло с начала года:

public static void main(String[] args) throws Exception

{

**Date** yearStartTime = **new** **Date**();

yearStartTime.setHours(0);

yearStartTime.setMinutes(0);

yearStartTime.setSeconds(0);

yearStartTime.setDate(1); // первое число

yearStartTime.setMonth(0); // месяц январь, нумерация для месяцев 0-11

**Date** currentTime = new **Date**();

long msTimeDistance = currentTime.getTime() - yearStartTime.getTime();

long msDay = 24 \* 60 \* 60 \* 1000; //сколько миллисекунд в одних сутках

int dayCount = (int) (msTimeDistance/msDay); //количество целых дней

System.out.println("Days from start of year: " + dayCount);

}

2

Задача

Java Syntax,  8 уровень,  4 лекция

Набираем код Ӏ Java Syntax: 8 уровень, 4 лекция

Java Syntax: 8 уровень, 4 лекция. Внимание! Объявляется набор кода на JavaRush. Для этого включите режим повышенной внимательности, расслабьте пальцы, читайте код и… набирайте его в соответствующем окошке. Набор кода — вовсе не бесполезное занятие, как может показаться на первый взгляд: благодаря ему новичок привыкает к синтаксису и запоминает его (современные IDE редко дают ему это сделать).

— Метод getTime() возвращает количество миллисекунд, которое хранится в объекте Date.

— Метод after() проверяет, что дата, у которой мы вызвали метод, идет после переданной в метод даты **after**.

— Методы getHours(), getMinutes(), getSeconds() возвращают количество часов, минут и секунд у объекта, у которого их вызвали.

— Более того, в последнем примере ты видишь, что можно повлиять на дату/время, сохраненное в объекте **Date**. Мы получаем текущее время и дату, а затем сбрасываем часы, минуты и секунды в 0. Так же устанавливаем месяц в Январь, а день месяца в 1. Теперь объект yearStartTime хранит дату и время 1 января текущего года 0 часов 0 минут 0 секунд.

— Потом мы снова получаем текущую дату currentTime, и вычисляем разницу в миллисекундах между двумя датами. Я про msTimeDistance.

— Затем делим msTimeDistance на количество миллисекунд в сутках и получаем полное количество дней, прошедшее с начала года до сегодняшнего дня!

— Вау! Вот это круто!