



# Задачи с подвохом

Для начала прочитайте поговорку:

ЛУЧШЕ СИНИЦА В  
В РУКАХ, ЧЕМ ЖУРАВЛЬ  
В НЕБЕ

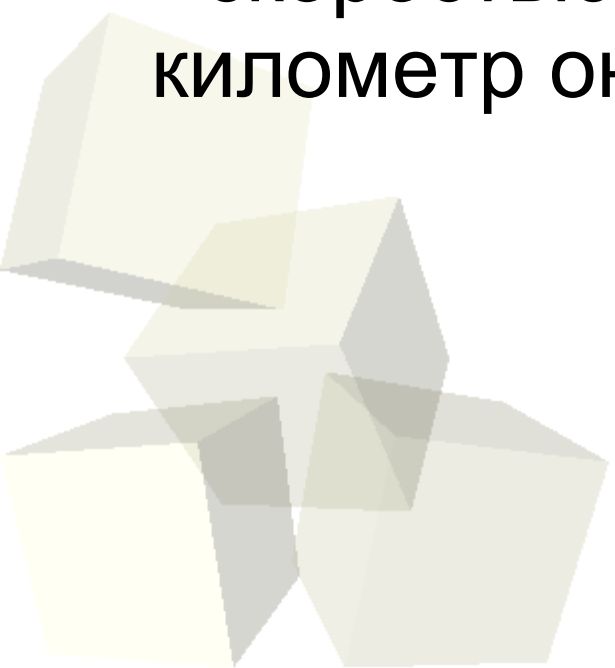
Что в ней необычного?





# Задачи с подвохом

Автомобиль едет со скоростью 60 км/ч. С какой скоростью он должен ехать, чтобы каждый километр он проходил на 1 минуту быстрее?





# Задачи с подвохом

Не так давно стал известен факт попадания в большинство водопроводных систем опасного химиката. Этот химикат бесцветный, безвкусный и не имеет запаха. Правительство не предприняло никаких попыток борьбы с этим заражением. Данный химикат называется дигидрогена монооксид (Dihydrogen monoxide).

Химикат используется для следующих целей:

- В производстве как растворитель и охладитель
- В ядерных реакторах
- В производстве пенопласта
- В огнетушителях
- В химических и биологических лабораториях
- В производстве пестицидов
- В искусственных пищевых добавках

Опасность этого вещества заключается в следующем:

- Химикат присутствует в составе кислотных дождей
- Вызывает эрозию почвы
- Ускоряет коррозию металлов и вредит большинству электроприборов при попадании внутрь
- Длительный контакт с химикатом в его твёрдой форме приводит к серьёзным повреждениям кожи человека
- Контакт с газообразной формой химиката приводит к сильным ожогам
- Вдыхание даже небольшого количества химиката грозит смертельным исходом
- Химикат обнаружен в злокачественных опухолях, нарывах, язвах и прочих болезненных изменениях тела
- Химикат развивает стойкую зависимость; жертвам при воздержании от потребления этого вещества грозит смерть в течение 168 часов

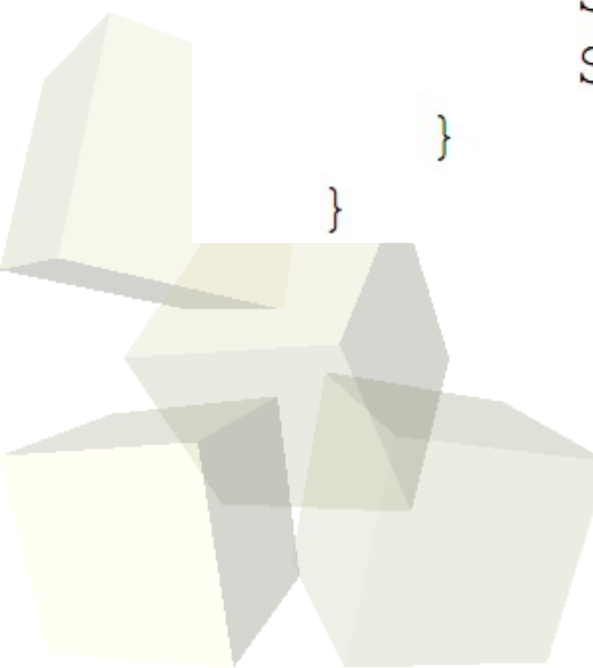
Несмотря на эти опасности, это вещество активно и безнаказанно используется в индустрии. Многие корпорации ежедневно получают тонны химиката через специально проложенные подземные трубопроводы. Люди, работающие с химикатом, как правило, не получают спецодежды и инструктажа. Отработанный химикат тоннами выливается в реки и моря.

О каком веществе идёт речь?



# Простое приложение

```
// пример # 1 : простое линейное приложение: First.java  
package chapt01;  
  
public class First {  
    public static void main(String[] args) {  
        // вывод строк  
        System.out.print("Мустанг ");  
        System.out.println("уже здесь!");  
    }  
}
```





# Запуск приложения

Простейший способ компиляции – вызов строчного компилятора из корневого каталога (в нем находится каталог chapt01):

```
javac chapt01/First.java
```

При успешной компиляции создается файл **FirstProgram.class**. Запустить этот виртуальный код можно с помощью интерпретатора Java:

```
java chapt01.First
```



# Простое приложение

*/\* пример # 2 : простое объектно-ориентированное приложение :*

*FirstProgram.java \*/*

```
package chapt01;
```

```
public class FirstProgram {
```

```
    public static void main(String[] args) {
```

```
        //объявление и создание объекта firstObject
```

```
        MustangLogic firstObject = new MustangLogic();
```

```
        //вызов метода, содержащего вывод строки
```

```
        firstObject.jumpMustang();
```

```
    }
```

```
}
```

*// пример # 3 : простой класс: MustangLogic*

```
class MustangLogic {
```

```
    public void jumpMustang() {// определение метода
```

```
        // вывод строки
```

```
        System.out.println("Мустанг уже здесь!");
```

```
    }
```

```
}
```



Здесь класс **FirstProgram** используется для того, чтобы определить метод **main()**, который запускается автоматически интерпретатором Java.

Тело метода **main()** содержит объявление объекта **MustangLogic firstObject = new MustangLogic();**

и вызов его метода **firstObject.jumpMustang();**



Метод **main()** содержит аргументы-параметры командной строки **String[] args**, представляющие массив строк, и является открытым (**public**) членом класса.





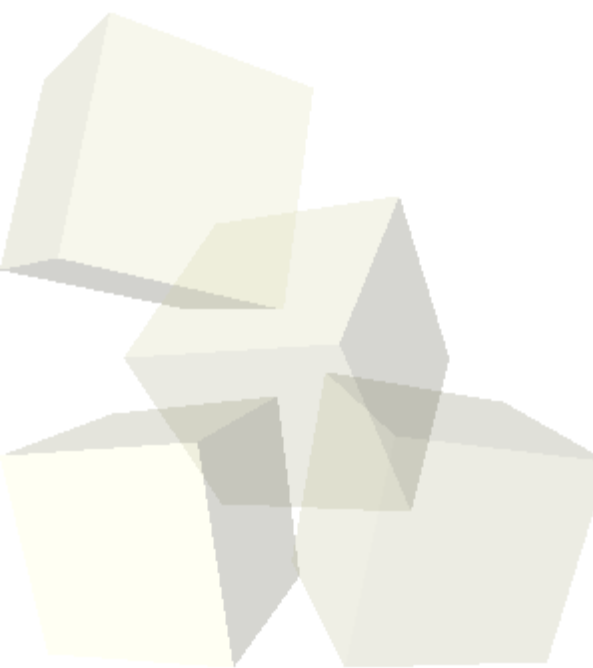


Количество аргументов определяется значением `args.length`.

*/\* пример # 4 : вывод аргументов командной строки: OutArgs.java \*/*

```
package chapt01;
```

```
public class OutArgs {  
    public static void main(String[] args) {  
        for (String str : args)  
            System.out.printf("Арг-> %s\n", str);  
    }  
}
```





Тот же результат был бы получен при использовании традиционного цикла

```
/* пример # 4 : вывод аргументов командной строки: OutArgs.java */  
package chapt01;  
  
public class OutArgs {  
    public static void main(String[] args) {  
        for (int i = 0; i < args.length; i++)  
            System.out.println("Арг-> " + args[i]);  
    }  
}
```

**java chapt01.OutArgs 2007 Mustang "Java SE 6"**

что приведет к выводу на консоль следующей информации:

**Арг-> 2007**

**Арг-> Mustang**

**Арг-> Java SE 6**