

Пример построения логики визулизатора

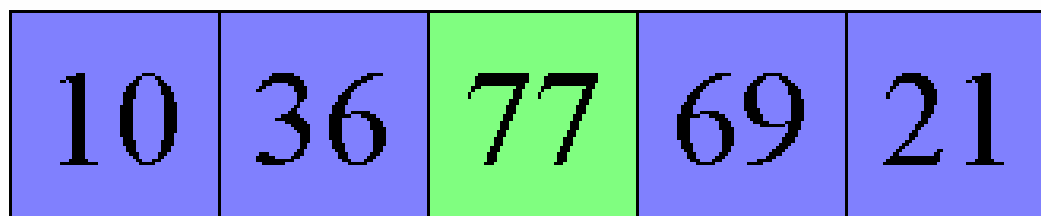
Георгий Корнеев
Анатолий Шальто

Кафедра технологий программирования
Санкт-Петербургского государственного университета
информационных технологий, механики и оптики

Построение логики визуализатора

Алгоритм поиска максимума в массиве
натуральных чисел

max = 36



77 больше текущего максимума (36)

<<	>>	Рестарт	Авто	<<	Задержка: 1000	>>	?
Случайно	Сохранить/Загрузить	<<	Элементов: 7	>>			

```
int max = 0;  
for (int i = 0; i < a.length; i++) {  
    if (max < a[i]) {  
        max = a[i];  
    }  
}
```

```
int max = 0;  
int i = 0;  
while (i < a.length) {  
    if (max < a[i]) {  
        max = a[i];  
    }  
    i++;  
}
```

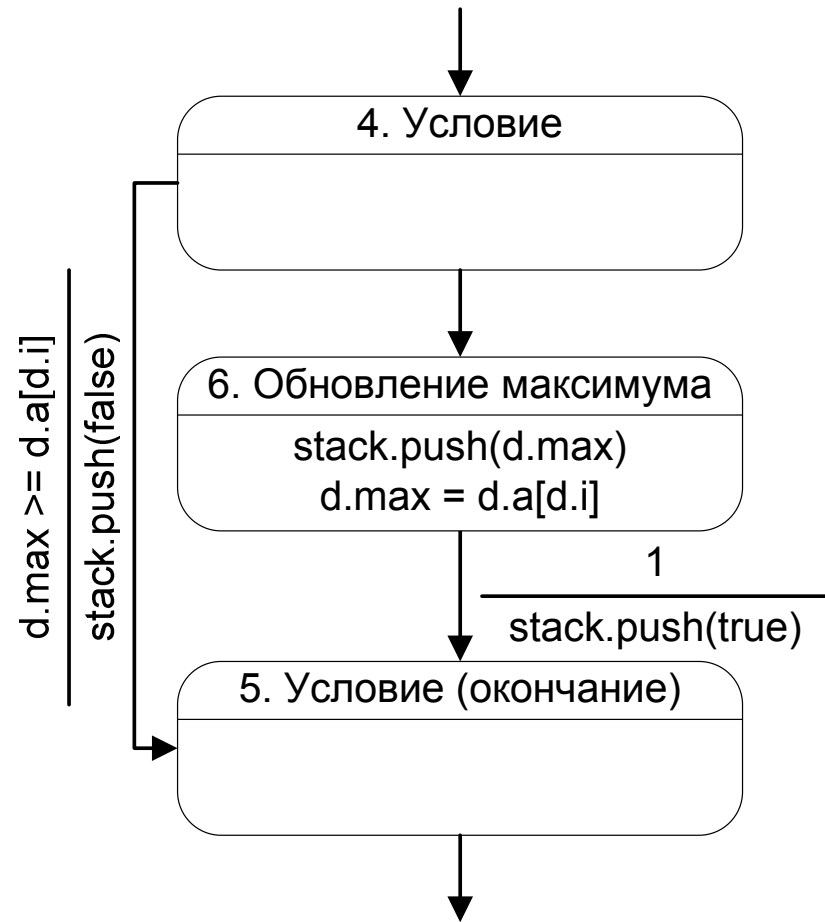
```
public final static class Data {  
    public int max;  
    public int a[];  
    public int Main_i;  
}
```

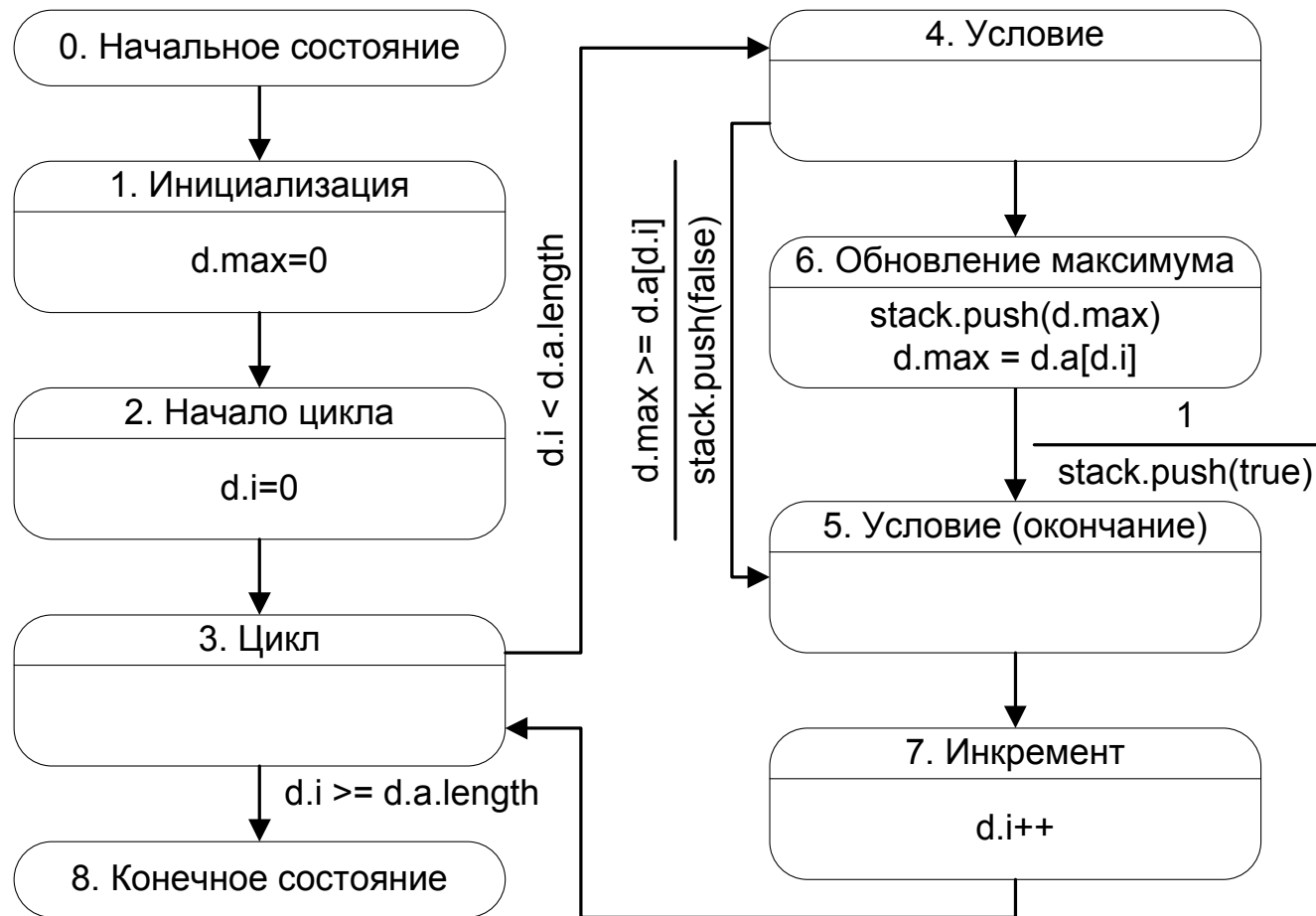
Реализация после выделения модели данных

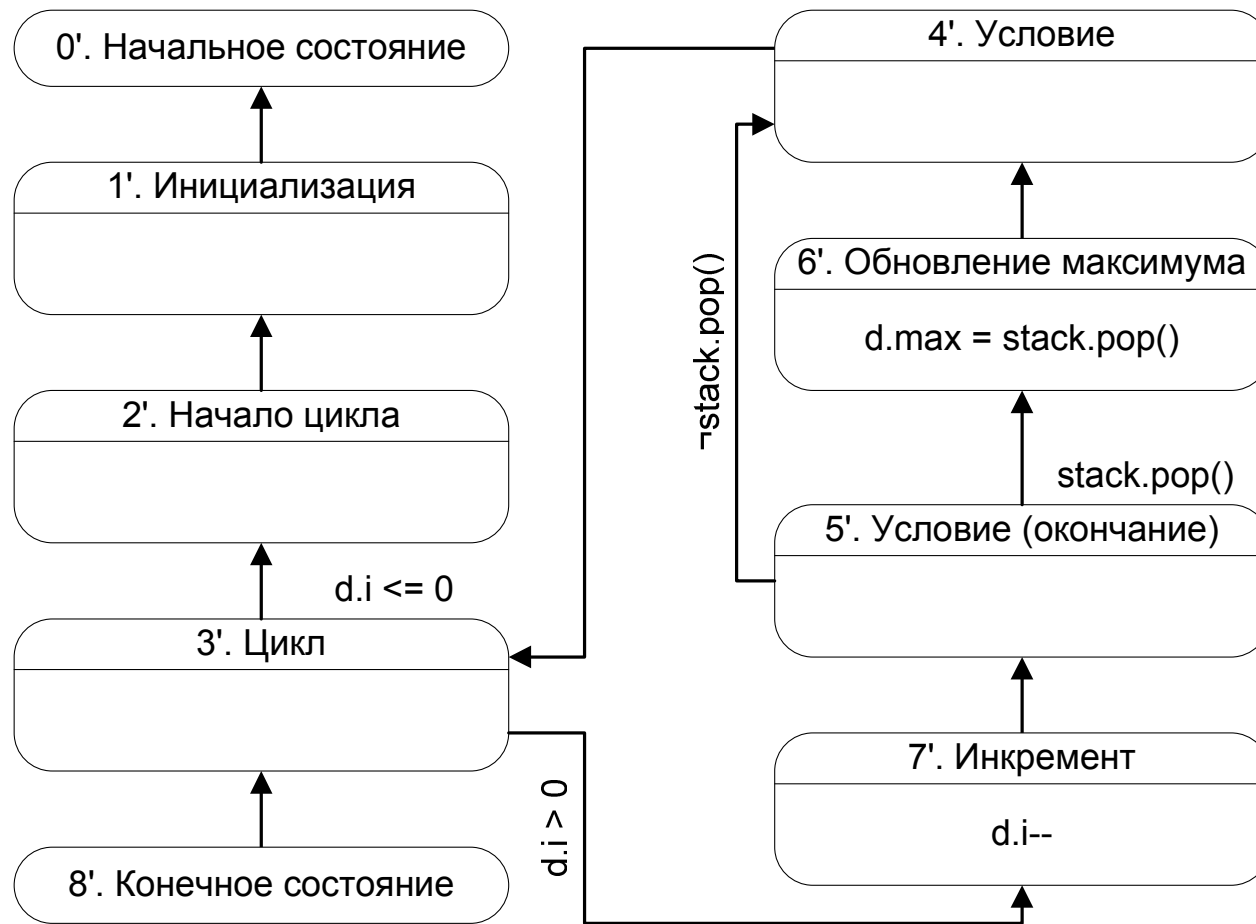
```
d.max = 0;  
d.i = 0;  
while (d.i < d.a.length) {  
    if (d.max < d.a[d.i]) {  
        d.max = d.a[d.i];  
    }  
    d.i++;  
}
```

Построение автомата (1)

```
if (d.max < d.a[d.i]) {  
    d.max = d.a[d.i];  
}
```







Фрагмент XML-описания логики визуализатора

```
if (d.max < d.a[d.i]) {...}
```

```
<if  
  id           = "Cond"  
  description  = "Условие"  
  test        = "@max < @a[@i]"  
  true-comment-ru = "{0} больше текущего максимума ({1})"  
  false-comment-ru = "{0} не больше текущего максимума ({1})"  
  comment-args  = "new Integer(@a[@i]), new Integer(@max)"  
>  
  <draw>  
    @visualizer.updateArray(@i, 1);  
  </draw>  
  ...  
</if>
```

Пример фрагмента кода (прямой автомат)

```
switch (state) {
  case 0: { // Начальное состояние
    state = 1; // Инициализация
    break;
  } case 1: { // Инициализация
    d.max = 0;
    state = 2; // Иниц. цикла
    break;
  } case 2: { // Иниц. цикла
    d.i = 0;
    state = 3; // Цикл
    break;
  } case 3: { // Цикл
    if (d.Main_i < d.a.length)
      state = 4; // Условие
    else state = END_STATE;
    break;
  }
}

case 4: { // Условие
  if (d.max < d.a[d.Main_i])
    state = 6; // Обновление
  else state = 5; // End if
  break;
} case 5: { // End if
  state = 7; // Increment
  break;
} case 6: { // Update
  d.max = d.a[d.Main_i];
  state = 5; // End if
  break;
} case 7: { // Increment
  d.Main_i++;
  state = 3; // Loop
  break;
}
```

Визуализатор алгоритма поиска максимума

- Автоматов 2
- Состояний 9
- Переходов 22

- XML-описание 87 строк
- Реализация автомата 326 строк

- Разработан метод преобразования программ в систему взаимодействующих конечных автоматов
- Разработанный метод реализован в пакете *Vizi* (<http://ctddev.ifmo.ru/vizi/>)
- Несколько примеров визуализаторов опубликованы на сайте <http://is.ifmo.ru>, раздел «Визуализаторы»

