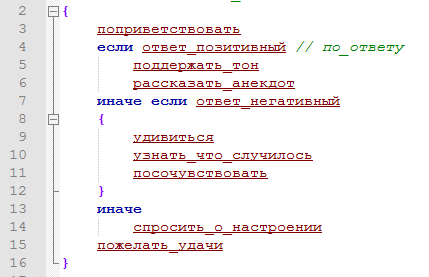
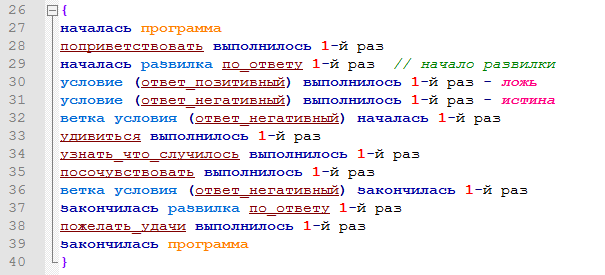
**Экран 1: ввод условий задачи**

Алгоритм со значениями управляющих условий:



Text area

Начальная трасса (можно оставить пустой):



Перейти к выполнению задачи

Text area

**Пример данных**

*// алгоритм alg*

**{**

**пока** не\_зелёный **->** ***истина*,*истина*,*ложь*** *// ожидание*

**{**

**если** цвет\_красный **->** ***истина*,*ложь*** *// по\_цвету*

**{**

ждать

**}**

**иначе** **если** цвет\_жёлтый **->** ***ложь***

**{**

приготовиться

**}**

**}**

**}**

*// alg трасса*

**{**

**началась** **программа**

**начался** **цикл** ожидание **1-**й раз

**условие** **цикла** **(**не\_зелёный**)** **выполнилось** **1-**й раз **-** ***истина***

**началась** **итерация** **1** **цикла** ожидание

**началась** **развилка** по\_цвету **1-**й раз

**условие** **развилки** **(**цвет\_красный**)** **выполнилось** **1-**й раз **-** ***истина***

**ветка** **условия** **развилки** **(**цвет\_красный**)** **началась** **1-**й раз

ждать **выполнилось** **1-**й раз

**ветка** **условия** **развилки** **(**цвет\_красный**)** **закончилась** **1-**й раз

**закончилась** **развилка** по\_цвету **1-**й раз

**закончилась** **итерация** **1** **цикла** ожидание

**условие** **цикла** **(**не\_зелёный**)** **выполнилось** **2-**й раз **-** ***истина***

**началась** **итерация** **2** **цикла** ожидание

**началась** **развилка** по\_цвету **2-**й раз

**условие** **развилки** **(**цвет\_красный**)** **выполнилось** **2-**й раз **-** ***ложь***

**условие** **развилки** **(**цвет\_жёлтый**)** **выполнилось** **1-**й раз **-** ***ложь***

**закончилась** **развилка** по\_цвету **2-**й раз

**закончилась** **итерация** **2** **цикла** ожидание

**условие** **цикла** **(**не\_зелёный**)** **выполнилось** **3-**й раз **-** ***ложь***

**закончился** **цикл** ожидание **1-**й раз

**закончилась** **программа**

**}**

Алгоритм и трасса выше представляются в виде JSON (см. файл *первая\_трасса.json*). В этом виде они отправляются на сервер для проверки.

**Экран 2: ввод ответа на задачу (трассы)**

Алгоритм и текущие значения управляющих условий:

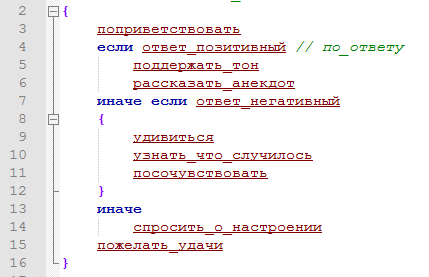
Выполнить условие услови

Закончить цикл

*Активные зоны (кружки) добавляют акты в трассу*

Начать цикл

Начать итерацию



Закончить ветку

Выполнить действие

Закончить итерацию

**пока** не\_зелёный *// ожидание*

**{**

**если** цвет\_красный *// по\_цвету*

**{**

ждать

**}**

**иначе** **если** цвет\_жёлтый

**{**

приготовиться

**}**

**}**

**->** ***истина***

**->** ***истина***

**->** ***ложь***

Трасса (построена нажатиями на цветные кружки):

**начался** **цикл** ожидание **1-**й раз

**началась** **итерация** **1** **цикла** ожидание

**условие** **цикла** **(**не\_зелёный**)** **выполнилось** **1-**й раз **-** ***истина***

**началась** **развилка** по\_цвету **1-**й раз

**условие** **развилки** **(**цвет\_красный**)** **выполнилось** **1-**й раз **-** ***истина***

**ветка** **условия** **развилки** **(**цвет\_красный**)** **началась** **1-**й раз

ждать **выполнилось** **1-**й раз

**ветка** **условия** **развилки** **(**цвет\_красный**)** **закончилась** **1-**й раз

**закончилась** **развилка** по\_цвету **1-**й раз

**закончилась** **итерация** **1** **цикла** ожидание

**условие** **цикла** **(**не\_зелёный**)** **выполнилось** **2-**й раз **-** ***истина***

**началась** **итерация** **2** **цикла** ожидание

**началась** **развилка** по\_цвету **2-**й раз

**условие** **развилки** **(**цвет\_красный**)** **выполнилось** **2-**й раз **-** ***ложь***

**условие** **развилки** **(**цвет\_жёлтый**)** **выполнилось** **1-**й раз **-** ***ложь***

**закончилась** **развилка** по\_цвету **2-**й раз

**закончилась** **итерация** **2** **цикла** ожидание

**условие** **цикла** **(**не\_зелёный**)** **выполнилось** **3-**й раз **-** ***ложь***

**закончился** **цикл** ожидание **1-**й раз

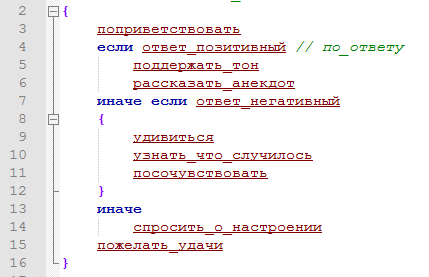
Проверить

Сбросить решение

**Экран 2: показ ошибок в ответе**

Ответ сервера приведён в файле «*ответ\_сервера.json*»

Алгоритм и текущие значения управляющих условий:



<Описание ошибки для этой строки> - tooltip

**->** ***истина***

**->** ***истина***

**->** ***ложь***

**пока** не\_зелёный *// ожидание*

**{**

**если** цвет\_красный *// по\_цвету*

**{**

ждать

**}**

**иначе** **если** цвет\_жёлтый

**{**

приготовиться

**}**

**}**

Трасса (ошибочные строки помечаются цветом):

Ошибки в трассе:

2: Акт начала развилки выполнился слишком рано, Нет проверки условия цикла

3: Во время выполнения цикла «ожидание» должно выполниться тело цикла, потому что условие цикла «не\_зелёный» истинно

4: Условие цикла не может выполняться в рамках акта итерации цикла, вложенного в акт цикла, потому что условие входит в цикл.

Проверить

Сбросить решение

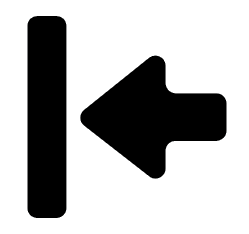
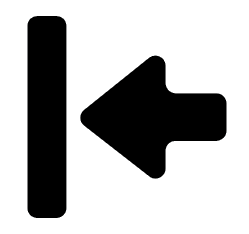
Во время выполнения цикла «ожидание» должно выполниться тело цикла, потому что условие цикла «не\_зелёный» истинно

Условие цикла не может выполняться в рамках акта итерации цикла, вложенного в акт цикла, потому что условие входит в цикл.

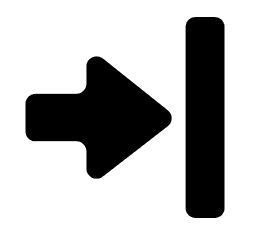
1. **начался** **цикл** ожидание **1-**й раз
2. **началась** **итерация** **1** **цикла** ожидание
3. **условие** **цикла** **(**не\_зелёный**)** **выполнилось** **1-**й раз **–** ***истина***
4. **началась** **развилка** по\_цвету **1-**й раз
5. **условие** **развилки** **(**цвет\_красный**)** **выполнилось** **1-**й раз **-** ***истина***
6. **ветка** **условия** **развилки** **(**цвет\_красный**)** **началась** **1-**й раз
7. ждать **выполнилось** **1-**й раз
8. **ветка** **условия** **развилки** **(**цвет\_красный**)** **закончилась** **1-**й раз
9. **закончилась** **развилка** по\_цвету **1-**й раз
10. **закончилась** **итерация** **1** **цикла** ожидание
11. **условие** **цикла** **(**не\_зелёный**)** **выполнилось** **2-**й раз **-** ***истина***
12. **началась** **итерация** **2** **цикла** ожидание
13. **началась** **развилка** по\_цвету **2-**й раз
14. **условие** **развилки** **(**цвет\_красный**)** **выполнилось** **2-**й раз **-** ***ложь***
15. **условие** **развилки** **(**цвет\_жёлтый**)** **выполнилось** **1-**й раз **-** ***ложь***
16. **закончилась** **развилка** по\_цвету **2-**й раз
17. **закончилась** **итерация** **2** **цикла** ожидание
18. **условие** **цикла** **(**не\_зелёный**)** **выполнилось** **3-**й раз **-** ***ложь***
19. **закончился** **цикл** ожидание **1-**й раз

**Пояснения по оформлению**

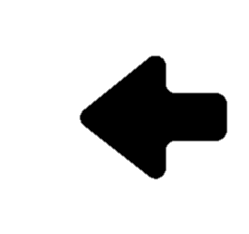
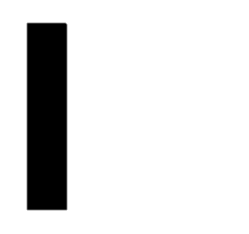
* Варианты иконок для создания актов – *начать*/*закончить*/*выполнить* (вместо кружков):

 (*начать*) (модифицированная в Word)

<https://www.flaticon.com/free-icon/rewind-to-beginning-with-left-arrow_56920?term=beginning&page=1&position=13>

 *(закончить)*

<https://www.flaticon.com/free-icon/right-arrow-symbol-pointing-a-vertical-line_57169?term=ending&page=1&position=67>

  
 (*выполнить*) (модифицированная в Word)

Над этими кнопками (при наведении) должно всплывать пояснение о том, какой акт будет добавлен в трассу.

* Ошибочные строки трассы должны подсвечиваться лёгким, но заметным красным фоном, при наведении должно всплывать пояснение (дублируя те, что выведены внизу, под трассой).

**Пояснения к данным**

Алгоритм:

* написан на псевдокоде, аналогичном по сути любому ЯП,
* с явным именованием составных действий (чтобы потом ссылаться) в комментарии справа от первой (главной) строки этой структуры (альтернатива, цикл, именованное следование),
* плюс значения управляющих условий («… **->** ***ложь***» ; «…**->** ***истина***» - здесь по одному значению на всю программу, т.к. больше не нужно, но в случае с циклами может быть и несколько значений, записанных через запятую),
* номера строк также должны присутствовать при JSON-описаниях каждого элемента алгоритма, по аналогии с актами трассы (в примере этого нет, но нужно добавлять).

Трасса:

* плоская последовательность актов (отступы в примере только для красоты)
* акты бывают двух видов:   
  **выполнилось** (атомарный акт)   
  и   
  **началось** / **закончилось** (составной акт, имеющий другие акты внутри себя, т.е. между началом и концом)
* каждый акт ссылается на элемент алгоритма по имени (в текстовой записи) и по ID в сериализованной форме (см. пример в JSON):

P.S.

Примеры алгоритмов и трасс (где они заключены в { } ) см. в файлах:

* github.com/den1s0v/c\_owl/master/blob/handcrafted\_traces??

Для форматирования цветом у нас применяются:

* Пользовательская подсветка в Notepad++ : github.com/den1s0v/c\_owl/master/blob/trace\_gen/NPP-UDL\_highlighting/Трасса\_npp.xml??
* «mode» для web-редактора CodeMirror, построенная на тех же принципах (см. в конце скрипта <http://vds84.server-1.biz:2020/iswc/demo/static/script.js>).