Turing’s Machine Realization Using Iterator.

II Part.

Realization of Turing’s Machine (Full Machine).

Formulated and created by Kosolapov Nikolai. 21.06.2021.



Table of contents:

* Сomparison of a simplified Turing’s machine with a full-fledged Turing’s machine.
* Structure and description: conditions system and logical aspects.
* Using realization for solving problems.
* Sources.

Comparison of a simplified Turing’s machine with a full-fledged Turing’s machine.

|  |  |
| --- | --- |
| Simplified version | Turing’s Machine |
| Not the original version of Turing’s Machine. | Original version, which is formulated by Alan Turing. |
| Simple structure. | Difficult structure. |
| Narrow Functionality | Wide Functionality |
|  |  |

The original Turing’s Machine has a condition system. So, we need to add system to implementation. Besides, I need to say, that in first part our simplified machine is not Turing’s machine. I created simplified version to show you basic processes and principles.

Structure and description: conditions system and logical aspects.

Condition of Turing’s machine – complex of instructions, intended for execution. Conditions system allows you to use multiple instructions for the same character.

Let’s illustrate structure of Turing’s machine with conditions system.

Instruction

Condition

Turing’s Machine

Instruction – command for reading head, which execute when the head reaches a certain symbol. Instruction - specific condition for certain symbol. Condition system has a number of properties:

1. Turing’s machine execute commands while current condition is not equal to q0(stop - condition).
2. If symbol is repeated, the instruction also repeated until the current character changes to a different one.
3. If the instruction is not defined for a particular symbol, the Turing’s machine stops working.

Example:

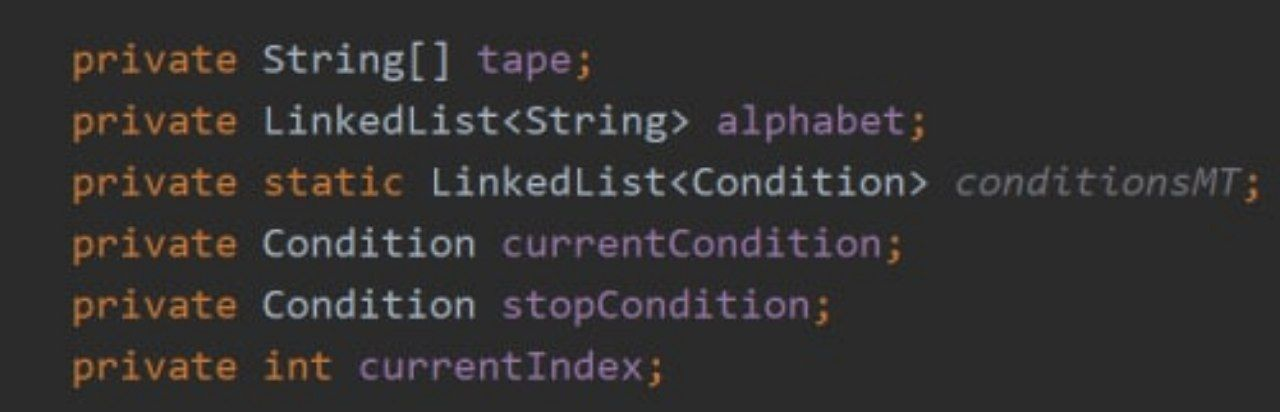
0 1 2 3 4 5 6 7 8 9

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a0 | a0 | a0 | a0 | C | V | C | V | C | a0 |

1. a0: q2a0L
2. C: q1CR
3. V: q1CR
4. a0: q0a0R
5. C: q2CL

1 condition 2 condition

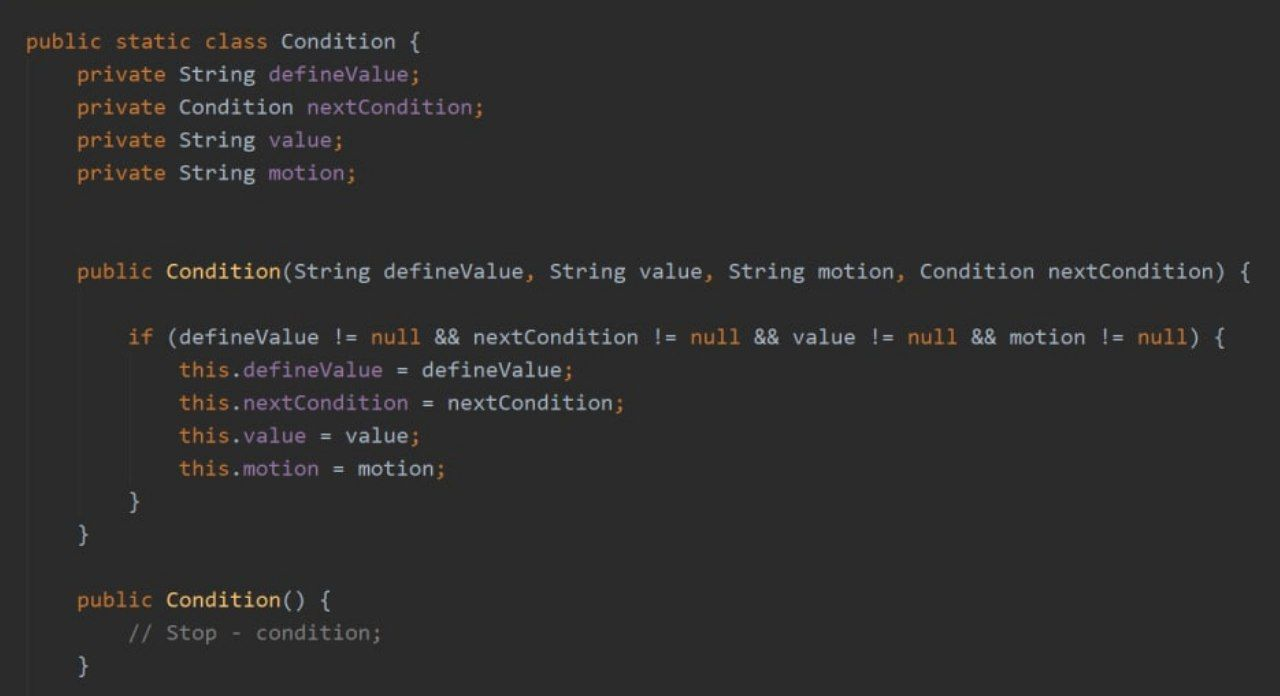
In my realization I designed following strucutere:



New components:

1. List of conditions.
2. Current condition;
3. Stop condition;

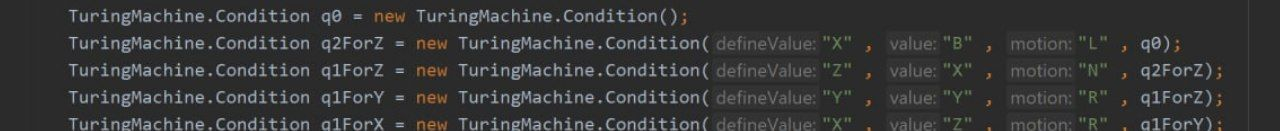
Other methods from first version and iterator remained.



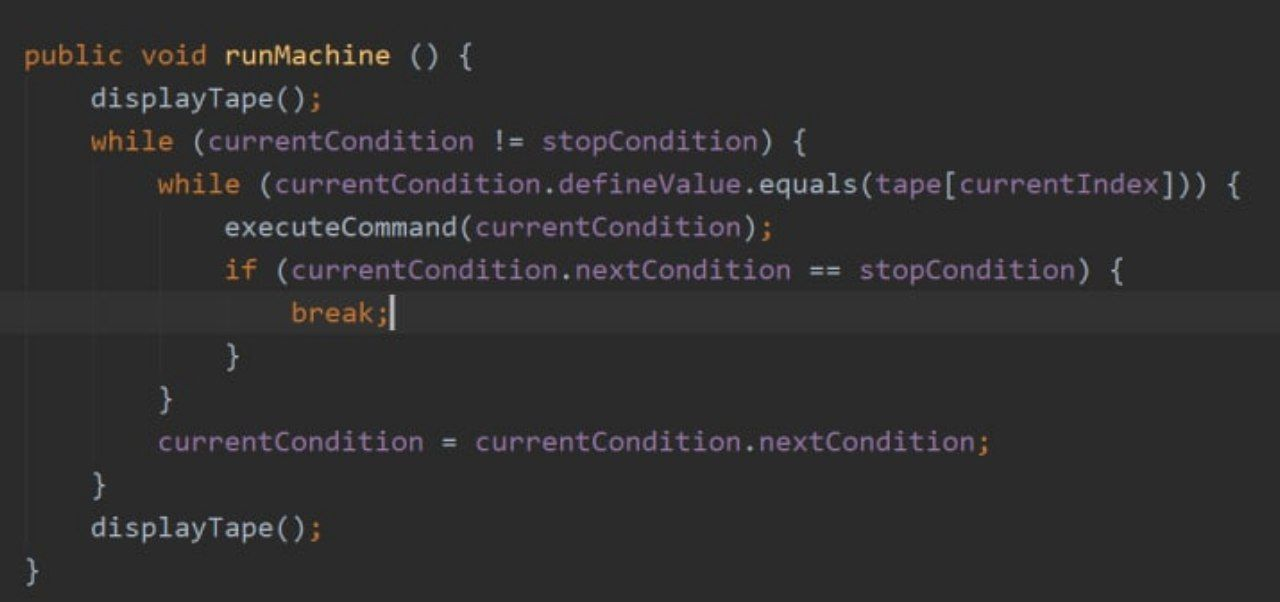
Empty constructor – constructor for stop-condition.

Params of instruction

Example:



How we can organize this difficult structure into working Turing’s Machine? Answer in following method:



1st property. 3rd property 2nd property

Using realization for solving problems.

Let’s use our Turing’s Machine to solve problem;

0 1 2 3 4 5 6 7 8

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | + | 1 | 1 | 1 |

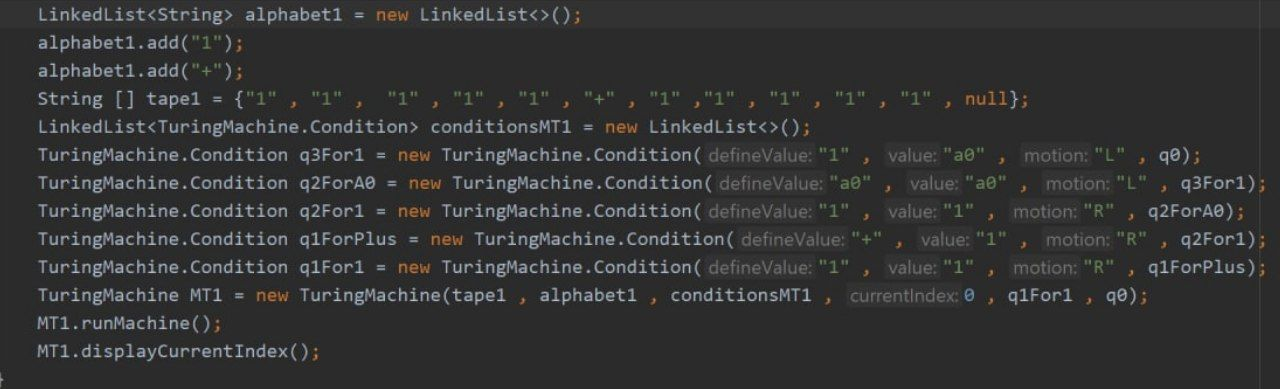
Let’s the numbers : 5 + 3.

(N number of consecutive units-record the number N)

We need to replace + with 1 and delete the last 1 in the record.

0 1 2 3 4 5 6 7 8

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | a0 |



Sources.

[First part of my research](https://github.com/KosolapovNikolai05/ResearchesArchive/commit/e41605451ba65e507618ba774ca80c080c533b2e)

I would like to thank Alan Turing for his great contribution to computer science