

# Procedural Automatic-Processing and K12 Education in Japan (1)

## — “Dolittle” Language Experiences —

Yasushi KUNO\*

2006.6.17

## Contents

- IPSJ's "Proposal 2005"
- "Dolittle" programming language
- "Dolittle" experiences

## 1 Proposal 2005

- Information Processing Society of Japan (IPSJ) --- largest academic society targeted to computer sciences / information technologies in Japan.
- ‘‘Proposal 2005 on Information and Information Processing Education in Japan’’ by IPSJ (2005.10.29).
  - Points to the problems in current Japanese IT environment (mainly on human resources) and proposes solution actions.

### 1.1 Backgrounds for the Proposal 2005

- Problems:
  - Japanese citizens lack appropriate IT understandings.
- Causes:
  - IT education in K12 education has only recently started.
  - Mainly targeted to IT literacy and IT society --- Principles of information processing are not taught much.
- We are NOT saying that IT literacy and IT society are unimportant --- they ARE certainly important!
- However, Japanese citizens ALSO need appropriate understanding of principle of computers.
  - This lack of understanding causes many problems --- e.g. System troubles in Tokyo securities trading market etc.
  - The only/quickest/easiest way to teach the principle: programming.

### 1.2 Procedural Automatic-Processing

- The second problem: many Japanese teachers are allergic to programming.
  - "Programming is too difficult to teach/learn; most students cannot understand the materials; consumes too many hours; ..."
  - So we have to investigate some strategies to make them accept our proposals...
- Our claim: "Procedural Automatic-Processing" is the heart of computers; we have to teach this to every students.
  - For example, spreadsheets, simulations, 3D renderings, and musical software include procedural automatic-processing.
  - However, we STILL think that programming is the most straight, quickest and easiest way to go.
- The process of "Procedural Automatic-Processing:"
  - (1) Recognize, describe and formalize the problem; then investigate for the solution.
  - (2) Construct and formulate the solution to an algorithmic and computer- executable form.
  - (3) Execute the solution, evaluate the outcomes and revert to the previous stages if not sufficient.
- Experience of "debugging" is very important to understand the nature of programming.
- Our proposal for K12 education...
  - All Japanese students should have experiences of procedural automatic-processing in EACH of elementary, junior-high, and high school.
  - Interested students should be able to take advanced classes at high school.
  - College entrance exam. should include subject on IT.

---

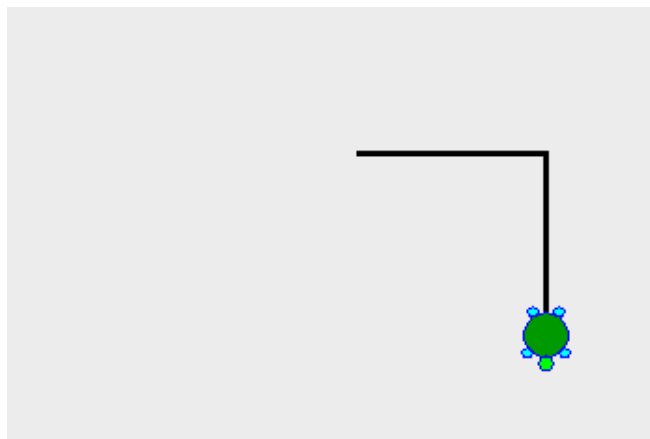
\*Graduate School of Business Sciences, University of Tsukuba, Tokyo, Japan

- CAUTION: "experience" is the point --- as in experiments in physics, chemistry, and biology.
  - With approx. 3 to 5 hours of experiences, students will have intuitive understandings of what computers can/cannot do.

## 2 "Dolittle" Programming Language

- To be successful in IT education, we need appropriate tool.
  - In the case of procedural automatic-processing, the tool should be a good "educational programming language (EPL)."
- Problems in traditional programming language classes:
  - Learn programming language syntax one-by-one (boring; time-consuming).
  - Unfriendly syntax (English-based, many symbols).
  - Resulting programs are unattractive (e.g. read two numbers; print their sum).
- Our EPL should be the contrary!
  - Simple and friendly syntax.
  - Students should be able to "walk by themselves" as quickly as possible.
  - Resulting program should be attractive (visual, multimedia, ...)
- Dolittle: a new EPL with the above properties.
  - Developed by Prof. Kanemune (Hitotsubashi Univ) and I.
- Use national (Japanese / Hangul) characters.
- Object-oriented (required for attractive outcome).
- Basic syntax: object! (param) (param) ... verb (param) ...verb.
  - Verbs comes last in an expression (natural for Japanese and Korean, might not be so for other countries...)
  - Program consumes smaller number of lines compared to C or Java (similar to ordinary text) --- more comprehensive (no scrolling).
- Lesson 1: instruction in predefined format.

```
Pen = Turtle ! create.
Pen ! 100 forward 90 rightturn 100 forward.
```

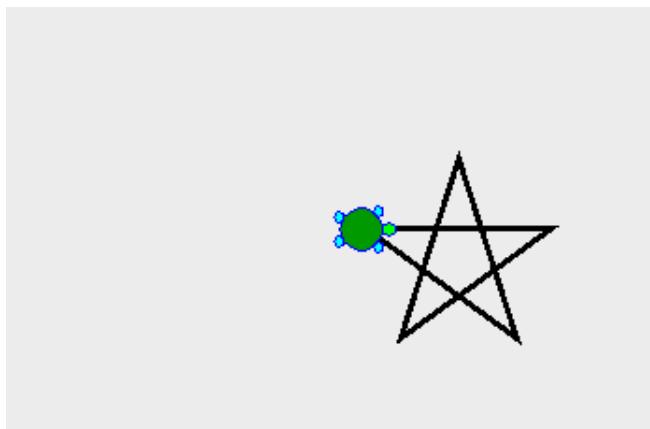


펜 = 거북 ! 만들다.  
 펜 ! 100 가고 90 우회전 100 가다.

ペン=タートル!作る。  
 ペン!100 歩く 90 右回り 100 歩く。|

- Students should type in the code and run the program. Then choose their original line drawings and craft their programs.
- Lesson 2: a loop in a small program can do lots of work.

```
Pen = Turtle ! create.
[Pen ! 100 forward 144 rightturn] ! 5 repeat.
```



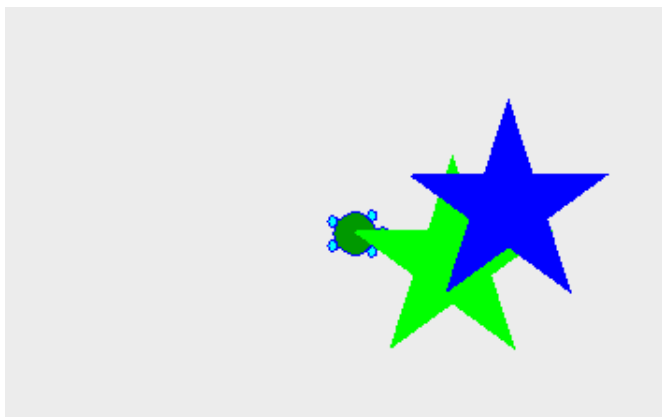
펜 = 거북 ! 만들다.  
 [ 펜 ! 100 가고 144 우회전 ] ! 5 반복한다.

ペン=タートル!作る。  
 「ペン!100 歩く 144 右回り」!5 繰り返す。

- Students should choose their original line drawings with repetition, and craft their programs.

- Lesson 3: anything a program handles is an "objects," and every object has their own properties and functionalities.

```
Pen = Turtle ! create.
[Pen ! 100 forward 144 rightturn] ! 5 repeat.
Star1 = Pen ! makefigure (green) paint.
Star2 = Star1 ! create 30 30 move (blue) paint.
```



펜 = 거북 ! 만든다.  
 [ 펜 ! 100 가고 144 우회전 ] ! 5 반복한다.  
 별1 = 펜 ! 도형만들고 (초록) 색칠하다.  
 별2 = 별1 ! 만들고 30 30 이동하고 (파랑) 색칠하다.

ペン=タートル!作る。  
 「ペン!100 歩く 144 右回り」!5 繰り返す。  
 星1=ペン!図形にする(緑)塗る。  
 星2=星1!作る 30 30 移動する(青)塗る。

- Students should design their original (painted) pictures, and craft their programs.
- In 5 hours, students will become capable of choosing their outcomes (problem statement), plan for programming code (solution), and debug their programs (evaluation/repetition) --- The core cycle of procedural automatic-processing.
- Other functionalities of Dolittle:
  - Animation (periodic execution).
  - Reactive GUI parts (buttons, menus, text fields...).
  - Musical outputs.
  - Distributed/networked programming.
  - Serial port I/O --- robot controls, sensors, ...

### 3 Experiences of Education Using "Dolittle"

- University of Tsukuba Attached High school

- Experimental Lecture (3 students, 3 hours)
- Drawing, timer (animation), objects and methods
- Students had many new findings on the principles of computers (screen pixels, operating systems, role of programming)
- Students enjoyed the classes.

- Kamata Junior High school, Mie prefecture.

- Mandatory classes (Technology subject, 11hours, 2nd grade, 132 students)
- Line drawings, figure objects, timer (animation), GUI parts.
- Evaluation by ordinary paper test for 3rd term --- approx. 80 to 90 percent have understood various concepts of Dolittle programming.
- Subjective enquiry --- as the curriculum proceeds, "difficulty" measure increased, but "enjoyable" measure also increased. Students have enjoyed challenge of programming.

- Voluntary class at Shimada city, Shizuoka prefecture.

- 17 nearby junior-high school students applied to the class. 4 hours: 1. turtle graphics, 2. ping-pong game (1 person), 3. networking, 4. kraft one's original network program.
- Students have learned what networks are and how they can be used from software.
- Many student chose text-based network communication program rather than network-based games (this was against teachers' expectation.) They might felt more interests in practical uses network communication.

- Other experiences...

- Robot control programming in Dolittle (high school, junior high school, elementary school)

- Students can write robot control programs in Dolittle, transfer the program to robot cars and run them --- plan, construct, debug process.

- Musical Programming in Dolittle (high school)

- Students experienced playing well-known (or original) melodies using Dolittle.

### 4 Summary

- Proposal 2005: IT Processing education is necessary.

- Procedural Automatic-Processing: effective tools for IT curriculum.
- Dolittle: educational programming language with simple, comprehensive syntax and 0-0.
- Dolittle experiences: many classes have successfully used Dolittle.

## 5 References

- IPSJ, Proposal 2005 on Information and Information Processing Education in Japan, 2005. (in Japanese)  
<http://www.ipsj.or.jp/12kyoiku/proposal-20051029.html>
- IPSJ, Comments on Accidents During late-2005 to early-2006 and their Relations to IT Education, 2006. (in Japanese)  
<http://www.ipsj.or.jp/12kyoiku/statement2006.html>
- Dae-Yong Kwon, Hye-Min Gil, Yong-Cheul Yeum, Seung-Wook Yoo, Susumu Kanemune, Yasushi Kuno, Won-Gyu Lee. Application and Evaluation of Object-Oriented Educational Programming Language ‘Dolittle’ for Computer Science Education in Secondary Education. The Journal of Korean Association of Computer Education, Vol.7, No.6, pp1-12, 2004.
- Susumu Kanemune, Yasushi Kuno. Dolittle: an object-oriented language for K12 education. EuroLogo 2005, Warszawa, Poland, pp144-153, 2005.
- Susumu Kanemune, Takako Nakatani, Rie Mitarai, Shingo Fukui, and Yasushi Kuno. Dolittle 2014 Experiences in Teaching Programming at K12 Schools. The Second International Conference on Creating, Connecting and Collaborating through Computing (C5), IEEE, pp.177-184, Kyoto, Japan, 2004. (invited paper)