

■ **Kanae Tsushima, Assistant Professor, Information Systems Architecture Science Research Division**

【Toward Reliable Programming Environments and Fun, Fruitful Programming Study】

During my first year of high school, I got really interested in the processes and procedures of chemistry experiments. After that, on the computer I had at home, I wrote and ran a small program, modified it and observed how the results changed. I wondered about the processes that took place inside a computer and decided to study informatics at college.

Toward the Reliability of the World with Types

When I was learning programming at college, I would encounter lots of errors related to ‘types.’ A type, in programming language, expresses whether something is a number or a character string. How these things are handled depends on the programming language.

For example, think of the example where you subtract a number Y from a number X and get the result Z, that is, $X - Y = Z$. If X and Y are numbers, then the result Z is also a number. So, X, Y, and Z are all of a numeric type. At the same time, it’s not possible to define a calculation in which a character string “spring” is subtracted from an integer “1.” “Spring” won’t go into Y to begin with, so we’d eliminate in advance programs that involve impossible calculations like subtracting a character string from an integer. By focusing only on types, we can accomplish things like that.

I like the reliability of the world with types. If there’s a type, “spring” can’t come from a place where you’d expect a whole number.

At the time, although there were systems in place to eliminate type errors in programs, they were inadequate. I thought I’d like to do something about this, so I chose this as my area of research.

I am still working on research on debugging type errors.

For More Fun, Fruitful Programming Study

Before, people who studied programming generally wanted to get a job in programming. Now, high school students study programming in their information classes, and there are also programming classes for elementary and middle school students and adults. The range of people involved in programming has expanded considerably.

Having said this, what to study and what to be done differ with age group and situation. What is the best way to teach programming? How can we make learning easier? Can we create computer systems to provide support in places where there no professionals really capable of helping people learn? Many issues come up when you begin to think in this way, and thinking about the solutions is quite fun. The joy of coming up with solutions is like the fun of defeating the bad guys in a video game.

The joy of programming lies in creating something you like and something fun on your own. To the extent that it’s possible, I think we need to try to eliminate the notion that it’s a lot of work or the attitude that I

have to study it though it's not very interesting. Even beginners should be able to concentrate in an enjoyable way on creating something that interests them. Understanding the basic concepts and workings of computers will help in the future even if you don't get a job that requires programming.

Through my research, I hope to improve Japan's overall information education system, including training for instructors.

(Article researched and organized by Yoshiko Miwa)