

Enseñar Programación

Avinash Kak
Purdue University
kak@purdue.edu

Inglés: publicado por primera vez: julio 2008; Revisado: abril 2016

Español: publicado por primera vez: septiembre 2018

DOS NOTAS IMPORTANTES

Disculpa de antemano por cualquier error. Por favor, lea la nota “Mi viaje al español” al final de este ensayo.

Si me envía un correo de electrónico sobre este ensayo, por favor escribe “Enseñar Programación” en la línea Asunto para pasar mi filtro de correo no deseado.

Resumen

Por lo general, a los profesores de las ciencias de la computación les gusta enseñar programación para usar lenguajes de programación más fácil porque creen que a sus estudiantes se lo gustan mejor. Pero ¿a intereses de largo plazo de los estudiantes les son consiguen por este enfoque? ¿Usar los lenguajes más fácil para enseñar programación no como enseñar musica con un banjo? Aun más fundamentalmente, ¿qué queremos decir por enseñar programación? Explorar estos temas es importante porque de el papel crítico desempeñado por el software ahora en todas las facetas de nuestras vidas.

1. Introducción

Así como los lenguas naturales se anclan el pensamiento humano, los lenguajes de programación se anclan nuestro comprensión del mundo de la informática. Por eso, si quieres entender el potencial y la limitación de los sistemas inteligentes de nuestro futuro, primero debes llegar a un acuerdo con el vocabulario de estos sistemas. Aunque hay muchos niveles de abstracción en este vocabulario, en su base está arraigada en los sustantivos, verbos, y otros calificadores de los lenguajes de programación.¹

¹ Este párrafo es desde el mi introducción a mi “*Objects Trilogy Project*” por <https://engineering.purdue.edu/kak/ObjectsTrilogy.html>. **El proyecto, que lleva diecisiete años, es ahora completo con el publicación de *Designing with Objects*.**

Considerando la importancia de los lenguajes de programación, se podría pensar que estos serían los centros de las ofertas de cursos de nuestras instituciones educativas. Desafortunadamente, ese no es caso frecuentemente. Enseñando un lenguaje de programación — así como debe enseñado — está pensado a ser demasiado gravoso por muchos.

Es verdad, por supuesto, cada institución que oferta la educación que está relacionada con la computadora enseña programación en una forma o otra. En un extremo del espectro se encuentran las instituciones educativas que empiezan los estudiantes con los lenguajes como C y luego construyen la experiencia en programación en una moda paso a paso a los lenguajes de programación de nivel superior y lenguajes de script. En el otro del espectro, tenemos las instituciones educativas que buscan el camino del éxito fácil y enseñan programación por un lenguaje que los profesores creen se encuentran más fácil por los estudiantes.

Son estas últimas instituciones que se han subido al carro de Java cuando se estalló en la escena hace varios años. Los profesores caían por la sintaxis directa de Java (sin punteros) y decidían que este lenguaje fue el mejor para introduciendo los estudiantes a programación y para explicando las ideas en las estructuras de datos, las estructuras de control, interfaces gráficas de usuario, etc. Creo que estas instituciones son responsables para muchos de los programadores que están aterrorizados tratar de resolver los problemas que frecuentemente consisten en fugas de memoria, desbordamiento de buffer, problemas de concurrencia, manejo de excepciones en tiempo de ejecución, problemas de optimización de rendimiento, depuración, probando, etc.

2. ¿Qué Significa Enseñar Programación?

Hasta hace unos 25 años, la respuesta a esta pregunta era clarísima. Durante el último cuarto de siglo, el mundo de computadores y programación se ha convertido tan diverso y variado que ya no es posible definir a que constituye las competencias básicas de programación. Hoy mientras casi todos con un licenciatura de ciencia o ingeniería — y también muchos con otros tipos de licenciaturas — aprender a programar los computadores en una forma u otra, es bastante común encontrar a los ingenieros y los científicos graduados que sus competencias de programación es limitada a usando los lenguajes de interpretado como Matlab o otros lenguajes similares. En caso el lector piense que esta situación es limitada a las escuelas que no enseñan las ciencias de computadores, se debe realizar que hay muchas las escuelas de este tipo donde la mayor parte de la enseñanza de programación es en Java o algún otro lenguaje fácil de usar.

Obviamente, no quiero decir en absoluto que los lenguajes interpretado como Matlab y lenguajes de programación como Java no merecen los puestos debajo el sol. Estas todos son las grandes lenguajes — estas han recibido muchos apoyos por los todos.

Para mí, la enseñanza de programación en una universidad significa, por una parte, enseñar cómo interactuar con una máquina de computación, y, por otra parte, cómo resolver los sorprendidos que el máquina puede echarte — *a pesar de que disponibilidad de los lenguajes que pueden aislarnos de muchos de los sorprendidos*. Aunque es buen sentido desarrollar un aplicación comercial con una lenguaje que viene con seguridad de tipo, comprobación del rango, y la recolección de elementos no utilizados, por estas cualidades exactamente que esas lenguajes son inadecuado como el medio primero para la instrucción programando.

Más específicamente, enseñando programación significa, por supuesto, enseñar los estructuras básicas de control que nos prestado por un lenguaje, y los estructuras de datos que podemos construir para realizar las varias problemas.

Por más que, enseñando programación, *en esencia*, también debe contener las artes y las ciencias de cómo implementar las estrategias eficientes para manejar las memorias, para crear los manipuladores de señal que permitiría un programa de ordenador interactuar con el sistema operativo, para entrada/salida de varios tipos de datos, para manejar los aspectos sincronización relacionadas de concurrencia, para control del proceso, etc.

Leyendo le párrafo anterior, muchos del mundo dirían que solamente los programadores de sistemas necesitarían a interactuar con la máquina en la nivel mencionada en la párrafo. Ellos añadirían que estos detalles no son importante para otros programadores. Quizás ese es verdad por una estudiante atrás ha licenciado y atrás estás empleado donde que solo necesitar lenguajes como Matlab. Pero cuando un estudiante está en un escuela para apreciar y disfrutar la programación, ese parte de la vida del estudiante no debería proteger de lo que se llama comúnmente los problemas del sistema.

Lo que he dije sobre que significa en esencia por enseñar programación adquiere importancia adicional en esta era cuando hasta las personas no tengan conocimientos técnicos puedan usar el vocabulario básico de informática. A pesar de la gente en general tal vez no sepa lo que es un programa de computación o cómo exactamente se hacer, muchos de estas personas usan confortablemente la jerga de ordenador como bits, bytes, las memorias de varios tipos, etc. Algunos de estos también supe los procesos en ejecución en sus ordenadores — las cosas que pueden ver cuando pulsan las tres teclas familiares CTL-ALT-DEL. Para estas personas, si una aplicación no parezca funcionar bien, ellos están confortablemente presionar CTL-ALT-DEL a ver el

listo de los procesos en el plántelas de sus ordenadores, identifican el proceso no funcional bien, y le borran. *Si se considera que ahora es habitual para la gente común a interactuar con el ordenador en lo que obviamente está el nivel sistema, ésta es una razón añadida porqué los programadores deben a entender las interacciones sistemas entre sus programas y el ordenador.*

La enseñanza de la programación significa también decir y mostrar cómo un programa mal escrito puede funcionar mal, y asegurar que los estudiantes se vuelven apasionado por desarrollar buenas habilidades informáticas. Pero eso no se puede hacer cuando el idioma de instrucción es un lenguaje que no le permite cometer errores.

3. Usar la Instrucción de la Musica Como la Metáfora para Enseñar Programación

La programación es mucho más que aprender solo la sintaxis de un lenguaje, de la misma manera que componer una pieza de música requiere mucho más que dominar las acordes que se toca en un instrumento musical. Aunque la ingeniería de software ahora ofrece varios enfoques para crear buen código fuente, sigue siendo fundamentalmente el caso que cuando creas cada entidad nueva (variables, estructuras de datos, etc.) en un programa, siempre estás pensando cómo esta interactuará con las entidades que creaste anteriormente y las entidades que crearás en el futuro. Esto es, cuando te fijas cada construcción en un programa, no puedes evitarse preocuparse sobre cómo esta armonizará con otras construcciones, esos ya en sitio e esos todavía no concebido.

4. Lo Que Dificulta la Enseñanza de la Programación

Si la enseñanza de programación solo significa explicar la sintaxis, el labor sería sencillo, pero también sería aburrida extraordinariamente (como cualquier estudiante quien que se ha enseñado la programación por un maestro no programador declararían). Para inculcar en los estudiantes la alegría de programación, el instructor debe demostrar los casos interesantes los que suceden cuando no adopción de buenas prácticas. Por ejemplo, se muestra fácilmente que hasta un ordenador poderoso se pone de rodillas por la creación deliberadamente de una pérdida de memoria como el máquina agota la memoria rápida. Del mismo modo, por desbordamiento del búfer, en la pila o en la montículo, se puede mostrar fácilmente cómo un trozo de código de programa

que está parecido inocente podría usar para dar respuestas no correctas.

Pero esta tipo de enseñanza se puede de realizar solo por quien es mucho participado con programación y desarrollar de software. Aunque las personas de esa especie son abundantes en la industria, son pocas y bastante distantes en las universidades. Es un hecho que muchas de los entornos educativos — especialmente en nuestras universidades de elite — están impulsados por las necesidades de los investigadores y muchos de ellos solo están interesados en impulsar del estado de la técnica en sus propias especialidades. Aunque estos investigadores quieren que sus estudiantes de posgrado escriban un buen código, ellos mismos no tienen el estómago para enseñar arte y ciencia a crear un buen software.

5. ¿Donde Nos Encontramos Hoy?

Hoy hay una dicotomía en las universidades: Por un lado, tenemos los jóvenes — la gran mayoría de los que no tienen ganas de estudiar para obtener los grados avanzados que se centran en la investigación — que tienen hambre de aprender las competencias de programación que se necesitan para competir a nivel mundial, y, por otro lado, una gran mayoría de los profesores quien no tienen tiempo ni ganas de impartir estas competencias a los estudiantes.

6. Por Último, una Cita ...

Considerando que este ensayo trata sobre la enseñanza de los lenguajes de la programación, la siguiente cita de la Lección 27 de mi popular “Lecture Notes on Computer and Network Security” parece apropiada:

“... en mi opinión, las personas que nos traen los lenguajes que se usan ampliamente son los deidades y profetas modernas. Obviamente, cientos de personas, si no miles, hacen contribuciones importantes a la maduración de esos lenguajes. No obstante, el crédito principal debe ir a las personas que fueron las primeras en concebirlas y luego lideraron sus evolución posterior. Este panteón incluye Dennis Ritchie para C, Bjarne Stroustrup para C++, James Gosling para Java, Larry Wall para Perl, Guido van Rossum para Python, Tim Berners-Lee para HTML, Rasmus Lerdorf para PHP, y varias otras.”

Acerca del autor

Avinash Kak es el autor de la serie de libros *Objects Trilogy*, con los tres libros de la serie publicados por John-Wiley. El primer libro, [Programming with Objects](#), se publicó en 2003, el segundo, [Scripting with Objects](#), en 2008, y el último, [Designing with Objects](#), en 2015. El último libro es un intento de explicar patrones de diseño más difíciles con la ayuda de las historias del mundo de Harry Potter. El autor es profesor de ingeniería eléctrica e informática en Purdue University.

Mi viaje al español

Hace algunos años, mi esposa, Stacey Smythe, tuvo que leer *Don Quijote* en español para uno de sus proyectos de historia del arte. Considerando que aprender un nuevo idioma es muy difícil y frustrante, decidí unirme a ella en el esfuerzo. Ese verano, ambos viajamos a Madrid para aprender español en la Universidad de Nebrija. En Madrid, me enamoré de tres cosas: (1) España, (2) la ciudad de Madrid; y (3) el idioma Español. Desde entonces, hemos viajado a Madrid todos los veranos para mejorar nuestro Español. Normalmente, asisto a una escuela (como Tandem Madrid) de dos semanas a un mes mientras Stacey trabaja en sus proyectos de historia del arte en los museos de Madrid. A pesar de este trabajo, sé que mi Español aún no es tan bueno como debe ser. Estoy seguro que este ensayo contiene muchos errores.