ارث‌بری در زبان شیءگرا و تناسب آن با اشتقاق کبری

یادم می‌آید که اوایل که من با این زبان‌ها آشنا شدم، پیاده‌کردن آن در اشتقاق کبری برایم جذاب شده بود. چون الآن در فضای کامپیوتر متعارف وقتی a، b، c ای الف، ب، ج می‌گویی‌د، این‌ها یک کارکتر است؛ یک کاراکتری که طبق جدول تبدیل خاص، در آن بایت‌ها ذخیره شده است. اما وقتی در زبان برنامه‌نویسی شیءگرا می‌آیی‌د - که اصالً سر و کار شما با غیر از شیء نیست و اگر بخواهی‌د آن را بشکنی‌د دیگر تمام شده و محو شده، مثل وقتی که مولکول آب را به هم بزنی‌د، آب می‌رود، وقتی شیء را هم به هم بزنی‌د، دیگر رفته - دیدم این برای اشتقاق کبری خیلی خوب است. چون وقتی شما «ب» ای «ر» می‌گویی‌د، سر و کار شما با یک نماد کاراکتری نیست، بلکه سر و کار شما با یک شیء است. از شیء چه کارهایی برمی‌آید؟ مثالً یکی از مهم‌ترین چیزهایی که در زبان شیءگرا هست، ارث‌بری است. یعنی یک کلاسی که والد است و ولد دارد؛ ولد از کلاس والد ارث می‌برد. این‌ها در فضای سامان‌دهی مطالب، چیز کمی نیست. لذا می‌بینی‌د وقتی شما «ب» می‌گویی‌د، اگر شیء باشد، بعداً چیزهایی که به «ب» می‌دهی‌د، آثار ارث‌بری «ب» را در تمام کلماتی که در آن‌ها «ب» به کار رفته، باید ببینی‌د، مگر در جایی که به صورت محلی(local) برایش حاکم بیاوری‌د و متغیرهای محلی بیاوری‌د تا ارث پدر را فعالاً محو کنی‌د. این‌ها بطور مفصّل کار شده و مطالب خوبی است.

مهم است که ما بفهمیم در یک فضایی، سر و کار ما با یک بیت نیست و اگرچه ماشین دارد کار با بیت‌ها را انجام می‌دهد، اما الآن خود ماشین وقتی دارد جابه‌جا می‌کند، در همان فضای جابه‌جا کردنش، دست‌به‌بندی کرده است، نه این‌که فقط پیش ناظر دست‌به‌بندی شده باشد، بلکه در حافظه، دست‌به‌بندی شده، گروه بندی شده. الآن اگر رَم را نگاه کنی‌د، می‌بینی‌د این‌ها در رَم دست‌به‌بندی شده است؛ شیء است، ولو ناظری این کار را کرده باشد. بعداً عرض می‌کنم. عرض اصلی من مانده است. بنابراین جزء لایتجزای شیء، کار مهمی داشت.

بازبینی #2
ایجاد شده 10 فوریه 2025 12:09:22 توسط ... .
به روزرسانی شده 10 فوریه 2025 12:37:02 توسط ... .