ترکیب بایت‌ها و تشکیل جزء لایتجزای جدید

ببینید اساس حرف ما همین است: ما در مسیر برنامه‌نویسی، این بایت‌ها را طوری ترکیب کنیم که به یک جزء لایتجزایی غیر از آن بایت‌ها و کاراکترها برسیم. بایت‌ها، خودشان کاراکترهای غیر متشابه الاجزاء هستند. یعنی یک بایت، نماینده a است، و یکی نماینده b است؛ یکی نماد پنج است و یکی نماد شش است؛ اینها غیر متشابه الاجزاء هستند. وقتی این بایت‌های غیر متشابه الاجزاء با هم ترکیب صورت می‌دهند، اینجا است که با ترکیب‌های مختلف این غیر متشابه الاجزاءها، ما متمکن هستیم به یک جزء لایتجزاهای جدید برسیم. این جزءها بسیار مهم هستند. یعنی شما بعداً به یک چیزی می‌رسید که نمی‌توانید دوباره آن را به بایت برگردانید و آن، دیگر جزء لایتجزای جدید است و فقط سر و کار شما با آن جزء لایتجزای جدید است.

خب این را عرض کنم؛ روالی که برای برنامه‌نویسی‌ها بوده، این بود: می‌گفتند برنامه‌های رویّه‌ای؛ یک دستور می‌داد و می‌گفت این کار را بکن و خود برنامه‌نویسی، تک‌تک باید اینها را کنار هم بچیند. یکی از پیشرفت‌های بسیار مهمی که در این برنامه‌نویسی شد، برنامه‌نویسی شیءگرا بود. شیءگرا یعنی به جای اینکه سر و کار برنامه‌نویسی با حافظه باشد و با این باشد که هر دستوری را تنظیم کند و بگوید چه کار بکن، آمدن این بایت‌هایی که قبلاً وُرُد(word) شده بود، وردهایی که سگمنت و رجیسترهایی بود که آن‌ها را اعمال می‌کردند؛ اول به زبان اسمبلی تبدیل می‌کردند و بعد به زبان ماشین تبدیل می‌کردند. بعد آمدن یک گام بسیار مهم برداشتند؛ گفتند ما در زبان برنامه‌نویسی یک جزء لایتجزی درست می‌کنیم و اسم آن را «object» به‌معنای شیء می‌گذاریم که این را نمی‌توانیم به هم بزنیم. یعنی ما در فضای حافظه، نه فقط فضای سطح بالا، بلکه حتی در خود حافظه و مدیریت آن، مجموعه‌ای از اطلاعات را یکجا به‌صورت جزء لایتجزی قرار می‌دهیم و می‌گوییم شیء. ابتدا که پیدا شد، زبان‌هایی مثل C بود که بعد ++C کردند، یعنی به آن ضمیمه کردند و مدام حالت شیءگرایی را به زور به آن دادند. بعد که دیدند عجب فکر عالی‌ای شد و خیلی پر فایده است، آمدن یک زبان‌های جدیدی مثل سی شارپ(#C) مایکروسافت ای پایتون را آوردند که اساسش بر شیءگرائی بود. یعنی اصلاً در این زبان پایه‌ای‌ترین چیزی که شما دارید، همین «object» است و اصلاً چیزی زیر آن ندارید؛ یعنی جزء لایتجزای اینجا، شیء می‌شود.

شیء چیست؟ شیء، دو چیز اساسی دارد؛ البته خیلی اجزاء دارد، اما اقل چیزی که شما از شیء سراغ دارید، این است که هر شیئی یک صفت دارد و یک کار دارد؛ یک کاری انجام می‌دهد و یک صفتی دارد. آن‌ها آمدن مجموعه اطلاعاتی که می‌خواستند به‌عنوان جزء لایتجزی کنار هم بگذارند، آن‌ها را در یک بخش حافظه به‌عنوان جزء لایتجزی جمع می‌کنند و می‌گویند این شیء است. شیء چیست؟ یک صفتی دارد و یک کار؛ «action» یا «Method» - اینها را کلمات مختلف به کار می‌برند – و یکی هم « Properties» است که به‌معنای صفات است. بنابراین در این زبان برنامه‌نویسی، شما با شیء سر و کار دارید. تفاوتش در چیست؟ من الآن یک مثال می‌زنم.

شما سه خط را بردارید و ترکیب کنید جوری که رأس‌های‌شان به هم بخورند، یک مثلث درست می‌شود. اگر این مثلث را اگر بخواهید با زبان‌های رویه‌ای رسم کنید، سر و کار شما با سه خط است

که مدام باید آن‌ها را کنار هم بگذاری. اگر رأس یک خط، بالا و پایین شد، باید خودتان با زحمت این سه تا را به هم وصل کنید تا یک شکل داشته باشید. نسخه‌های اولیه نرم‌افزار word هم همینطور بودند و وقتی می‌خواستید شکل بکشید همین کار را می‌کردید. اما وقتی شیءگرائی خیلی پررنگ شد، حالا می‌گویید یک شکل مثلث در این سند برای من ایجاد کن؛ یک مثلث با سه رأس می‌آید. بعد آن را می‌گیری و بزرگ می‌کنید و کوچک می‌کنید. این یعنی الآن سر و کار شما با یک شیء است به نام مثلث، نه با سه خط که مدام با زحمت باید آن‌ها را به هم وصل کنید؛ آن روی‌ها می‌شود. ولی در اینجا سر و کار شما با یک شیء است و هیچ وقت سر این سه خط از هم جدا نمی‌شوند و هر جوری هم این طرف و آن طرف بکشید -مهم‌ترین عرض من این است- طبق ضوابط فرمول‌های نسبت‌های مثلثاتی جلو می‌رود و الآن دیگر دست شما نیست و دست برنامه هم نیست؛ وقتی شما در حافظه، یک چیزی را ایجاد کردید به نام مثلث، کلاسی دارید به نام مثلث که یک نمونه‌های از آن را ایجاد می‌کنید؛ الآن دیگر دست شما نیست که زاویه آن را کم کنید یا زیاد کنید. اگر این زاویه را زیاد کنید، دو زاویه دیگر کم می‌شوند و اگر این ضلع را زیاد کنید، آن دو تا ضلع تغییر می‌کند. چرا؟ به‌خاطر این‌که وقتی شیء شد، تابع یک سری مطالب نفس الامری است و حالا دیگر حتی تابع خود حافظه هم نیست. توضیحات بیشترش را بعداً عرض می‌کنم.

بنابراین به یک اصل بسیار مهم می‌رسیم. وقتی سر و کار ما از بیت‌ها به بایت رسید و به معنا رسید، و بعد به شیء رسیدیم، حالا دیگر اختیار از دست ما درمی‌رود. وقتی سر و کار ما با یک جزء لایتجزای شیء است، باید ببینیم خصوصیات و کار آن شیء چیست. مثلثی که شما به‌عنوان یک شیء نمونه آن را ایجاد می‌کنید، نسبت‌های مثلثاتی و جدول‌های اتان‌ژانت و سینوس بر آن حاکم است و نمی‌توانید همینطور بگویید من باید به برنامه یک مثلثی را بدهم که مجموع زوایای آن 180 درجه نباشد و من می‌گویم به این خط که چطور باشد. نه، دیگر نمی‌تواند چنین باشد.

بازبینی #2
ایجاد شده 9 ژانویه 2025 08:09:19 ... .
به روزرسانی شده 9 ژانویه 2025 09:30:06 ... .