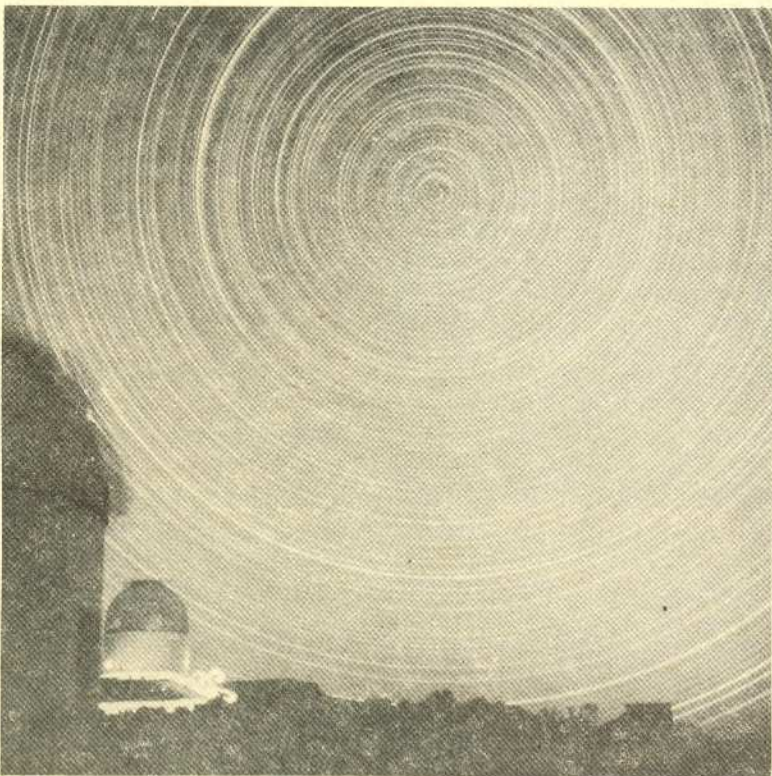


Joanna UDALSKA

Czy można fotografować niebo zatrzymanym teleskopem? Oczywiście można, lecz na skutek ruchu dziennego sfery niebieskiej gwiazdy na zdjęciu wyglądają jak łuki lub kreski. Każdy pewnie widział fotografie okolic bieguna nieba wykonane nieruchomym aparatem fotograficznym (teleskop fotografujący niebo to w końcu duży aparat fotograficzny) ilustrujące pozorny ruch sfery niebieskiej. Czy można jednak nieruchomym teleskopem uzyskać „normalne” obrazy nieba z gwiazdami w postaci punktów? Tu, oczywiście, nasuwającą się odpowiedzią jest: nie. A jednak! Najpierw dwa słowa o tym, jak we współczesnej astronomii fotografujemy niebo.

Gwałtowny rozwój elektroniki spowodował, że coraz rzadziej astronomowie używają dziś kliszy fotograficznej jako detektora. Nawet najlepsze klisze mają tzw. wydajność kwantową (procent fotonów zamienianych na elektrony) rzędu kilku dziesiątych procenta, a ich liniowość (zakres jasności, w którym zaczerpnięcie kliszy jest proporcjonalne do jasności obserwowanego obiektu) jest ograniczona do zaledwie kilku wielkości gwiazdowych. Najpowszechniejszym detektorem stosowanym obecnie do obrazowania nieba jest *Charge Coupled Device*, zwane w skrócie CCD. Jest to dwuwymiarowy detektor składający się z kilkuset tysięcy fotoczułych elementów umieszczonych na układzie scalonym w postaci siatki kilkaset na kilkaset rzędów i kolumn (patrz *Delta* 7/1986). Poszczególne elementy mają typowy rozmiar około $20 \mu\text{m}$, są więc mniejsze lub porównywalne z ziarnem kliszy. Podczas naświetlania CCD światło padające na poszczególne fotoelementy wybija fotoelektrony uwięzione w ich obszarze. Po zakończeniu ekspozycji następuje odczytywanie układu.



Niebo sfotografowane w chilijskim obserwatorium Las Campanas. Oczywiście, jest to zdjęcie wykonane nieruchomym aparatem. Dlatego też gwiazdy trochę się poruszyły (czas ekspozycji: 9 godzin; zdjęcie wykonał R. F. Garrison).

O ewolucji języków programowania

Ryszard KUBIAK,
Stefan ZEMKE

Jesteśmy wszyscy świadkami dynamicznego rozwoju technologii komputerowej. Postępuje miniaturyzacja sprzętu, a jednocześnie wzrastają jego możliwości. Jednak zmiany następują nie tylko w tej dziedzinie. Przemianom ulega również oprogramowanie, w tym języki programowania.

Bogactwo języków programowania

Jednym ze skutków czterdziestoletniego rozwoju informatyki jest powstawanie ogromnej liczby języków programowania. Proces ten rządzi się podobnymi prawami, co rozwój języków naturalnych. Bywają języki bardzo rozpowszechnione oraz takie, które są używane tylko w małych społecznościach specjalistów z danej dziedziny. Niektóre języki wymierają, a w ich miejsce pojawiają się nowe, o wyższych walorach użytkowych. Języki potrafią żyć własnym życiem, wzbogacając się o coraz to nowe formy dialektowe.

W niniejszym szkicu spróbujemy Czytelnikowi przedstawić zaznaczające się główne tendencje rozwoju języków programowania. Będą nas interesować jednak tylko tak zwane języki ogólnego zastosowania. W takich językach można pisać programy dotyczące zagadnień inżynierskich, baz danych, redagowania tekstów. Można też pisać programy rozpoznające i interpretujące napisy w danym języku, tak zwane interpretery lub kompilatory. Zostawimy natomiast na boku języki wąsko specjalizowane, przeznaczone do sterowania konkretnymi systemami informatycznymi. W tej grupie znajdzie się na przykład język porozumiewania się z systemem rezerwacji miejsc lotniczych albo język opisu rysunków technicznych w systemie komputerowego wspomagania projektowania.

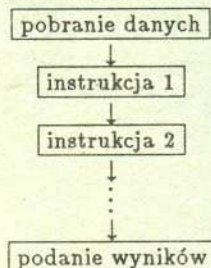
Języki programowania ogólnego zastosowania można podzielić na trzy grupy:

- języki imperatywne,
zwane też instrukcyjnymi;
- języki aplikatywne,
zwane też funkcyjnymi;
- języki deklaratywne,
zwane też logicznymi.

To, co odróżnia te trzy grupy języków, nie sprowadza się wyłącznie do reguł gramatycznych. Zasadniczym wyróżnikiem jest odmienny w każdym przypadku sposób myślenia programisty o stworzonym przez niego programie. W pewnym sensie, mamy tu analogię do różnic w sposobie myślenia o otaczającym nas świecie, gdy posługujemy się językiem polskim, angielskim czy chińskim. I aby dobrze władać danym językiem, nie wystarczy tylko znajomość obowiązujących w nim reguł gramatycznych. Trzeba jeszcze umieć w tym języku myśleć.

Myślenie w kategoriach instrukcji

Programowanie w języku imperatywnym polega na wypisywaniu ciągów instrukcji, które następnie przekazujemy komputerowi do wykonania.



Programista myśli o swoim zadaniu tak: gdy program otrzyma takie a takie dane, to należy kolejno wykonać na tych danych takie a takie operacje i w końcu zaprezentować wyniki. Świat w pamięci komputera zmienia się pod wpływem wykonywanych instrukcji. Programista musi w pełni panować nad dynamiką tych zmian.

Myślenie tego rodzaju jest nieodłącznie związane z budową i działaniem współczesnych komputerów. To właśnie komputer wykonuje kolejno zleczone mu instrukcje i wydaje się naturalnym programować w terminach dla niego podstawowych. Jest to zasadniczy powód, dla którego właśnie języki typu imperatywnego są najpowszechniej używane. Ugruntowany jest też pogląd, że dają one najpełniejsze możliwości efektywnego sterowania zasobami komputera: pamięcią, ekranem, drukarką, dyskami, itp.

Najpierw przez przyłożenie odpowiedniego napięcia pierwsza kolumna przesuwana jest do dodatkowej kolumny zerowej i wszystkie pozostałe przesuwają się o jedno miejsce w tym samym kierunku. Kolumna zerowa składa się, oczywiście, z tylu elementów, ile jest rzędów. Następnie kolejne komórki kolumny zerowej przesuwane są (również przez przykładanie odpowiedniego napięcia) do rejestru odczytującego. Napięcie zmierzone w rejestrze odczytu jest zamieniane na wartość cyfrową i zapisywane przez sterujący komputer w odpowiednim elemencie pamięci komputera. Po pomiarze do rejestru odczytującego przesuwana jest następna komórka itd., aż do odczytania całej kolumny. Po zakończeniu odczytu pierwszej kolumny w jej miejsce przesuwana jest następna (czyli druga) i proces jest powtarzany – aż do zakończenia odczytu ostatniej kolumny.

Ogromna wydajność kwantowa elementów CCD, sięgająca czasem nawet 80%, sprawia, że rejestruje ono niemal całe światło obiektów niebieskich. Duża pojemność elementów CCD, sięgająca kilkuset tysięcy elektronów, powoduje, że układy te są liniowe w zakresie wielu wielkości gwiazdowych, co pozwala mierzyć jasności gwiazd zarówno słabych, jak i jasnych na tym samym obrazie. Są jednak i problemy. Główny to technologiczny – problem wykonania dużych układów mających wszystkie elementy o podobnych parametrach. Aktualnie największe działające układy zawierają 2048 na 2048 elementów skupionych na płytce o rozmiarach około 4 cm na 4 cm i wydaje się, że dość trudno będzie wytworzyć większe. Taki układ pokrywa (w zależności od teleskopu, z którym współpracuje) obszar nieba zaledwie od kilku do kilkunastu minut łuku. Do wykonywania obrazów większych obszarów pozostają więc ciągle jeszcze klisze fotograficzne (mogą mieć nawet do 0,5 na 0,5 m).

Wróćmy teraz do problemu fotografowania nieba zatrzymanym teleskopem. Oczywiście, jeśli użyjemy kliszy fotograficznej, uzyskamy obrazy w postaci kresiek. Co będzie jednak, gdy jako detektora użyjemy CCD i ustawimy go tak, by rzędy były równoległe do kierunku ruchu dziennego nieba (oczywiście, ze względu na małe rozmiary CCD możemy przyjąć, że łuki są liniami prostymi)? Jeśli zastosujemy teraz standardowy sposób odczytu: ekspozycja, odczyt – to efekt będzie identyczny jak w przypadku kliszy. Zastanówmy się jednak, co uzyskamy, gdy będziemy odczytywać układ w sposób ciągle i dobierzemy czas odczytu kolumny równy dokładnie czasowi przesunięcia się gwiazdy do sąsiedniej kolumny na skutek ruchu sfery niebieskiej. Załóżmy dla ułatwienia, że obraz gwiazdy zajmuje tylko jeden element (w rzeczywistości na skutek drgań atmosfery i niedoskonałości optyki teleskopów z reguły zajmuje on kilka elementów).

Światło gwiazdy dotarłszy do układu pada na któryś z elementów ostatniej kolumny. W międzyczasie odczytywana jest kolumna zerowa. Po zakończeniu ostatnia kolumna przesuwana jest na pozycję przedostatnią, w niej więc znajdują się teraz fotoelektrony wybite przez światło gwiazdy. Ale ponieważ odpowiednio dobraliśmy czas odczytu, również światło gwiazdy pada teraz na element kolumny przedostatniej (oczywiście, rząd jest ten sam co poprzednio, bo ustawiliśmy układ równoległe rzędami do ruchu dziennego). Fotoelektrony akumulują się teraz z wybitymi poprzednio. Po odczycie kolejnej kolumny interesujący nas element przesuwany jest na miejsce przed-przedostatnie, gwiazda przesuwa się również tam i tak dalej aż do osiągnięcia kolumny pierwszej. Po dojściu do kolumny pierwszej element zawiera wszystkie fotoelektrony wybite wzdłuż całego rzędu. Odczytując tę kolumnę dostajemy punktowy obraz gwiazdy o czasie ekspozycji równym czasowi przechodzenia gwiazdy przez układ.

Nic więc prostszego, jak odczytywać układ bez przerwy. Zamiast obrazu o wielkości kilka na kilka minut łuku, jak przy klasycznym zastosowaniu

CCD, uzyskamy obraz paska o szerokości kilku minut na ... no właśnie, to zależy od tego, jak długo będziemy odczytywać. Może to być stopień, kilka czy kilkanaście. Technika ta daje więc możliwości uzyskiwania obrazów CCD dużych obszarów nieba, czyli tego, co w zasadzie do dzisiaj było domeną klisz fotograficznych. A zalety są ogromne. Przede wszystkim duża czułość CCD powoduje, że przy obserwacjach zatrzymanym teleskopem o średnicy lustra 1 metr rejestruje się obiekty słabsze niż na „najgłębszych” przeglądach fotograficznych. Ponadto istnieje zawsze możliwość regulacji czasu ekspozycji przez wydłużenie czasu przechodzenia gwiazdy przez układ, np. dzięki wprowadzeniu teleskopu w ruch wolniejszy od ruchu sfery niebieskiej. Zauważmy też, że czas ekspozycji jest, oczywiście, funkcją deklinacji. Uzyskane w ten sposób dane mają znakomitą precyzję zarówno astrometryczną, jak i fotometryczną; mają więc znacznie większą wartość niż zwykle fotografie.

Technika ta, zwana z angielska techniką skanowania, została z sukcesem zastosowana ostatnio przez astronomów amerykańskich, Paula Schechtera i Steva Shectmana, w obserwatorium Las Campanas w Chile.

Wykorzystali oni do tego najnowszy układ CCD o wielkości 2048 na 2048 elementów. Oczywiście, jak to zwykle bywa w przypadku pionierskich przedsięwzięć, nie obyło się bez problemów technicznych. Okazało się po pierwsze, że czas odczytu kolumny tak dużego układu jest zbyt długi (każda taka kolumna to przecież 2048 elementów), zaszła więc konieczność stosowania tzw. superelementów, składających się z czterech (2 na 2) podstawowych komórek. CCD miał więc efektywną wielkość 1024 na 1024 superelementów. Innym ograniczeniem była ogromna ilość danych spływających z układu – jedna kolumna w tej konfiguracji zawiera 1024 razy 2 bajty. Obliczmy: przy światłosile 1 : 7 teleskopu o średnicy 1 m, 1 mm w ognisku teleskopu odpowiada około 30 sekundom łuku. Jeśli obserwujemy gwiazdę na równiku niebieskim (najszybciej poruszającą się ze względu na ruch sfery niebieskiej) – w ciągu 1 sekundy przemieszcza się ona o 15 sekund łuku, czyli o pół milimetra na układzie CCD. Pół milimetra to 0,5/0,02, czyli 25 kolumn oryginalnych lub 0,5/0,04, czyli 12,5 kolumn superelementów. Tak więc na sekundę odczytywano 25 kB danych. W tej sytuacji problemem staje się ich magazynowanie. Nie udało się, niestety, zapisywać ich bezpośrednio na ostatnim cudzie techniki – kasetkach EXABYTE. Kasetki te mają pojemność 2,2 GB, a więc jedna wystarczyłaby na rejestrację danych z całej nocy! Z tej przyczyny ograniczono skany

do kilkunastu minut i dopiero po zakończeniu ekspozycji wyniki zapisywano na kasetkach. Ogólnie jednak obserwacje zakończyły się pełnym sukcesem i pozwoliły zebrać dane przeglądowe nieba z wielu obszarów.

Paul Schechter twierdzi, że już wkrótce będziemy mieli cyfrowe przeglądy nieba. I rzeczywiście, choć pewnie nieprędko zostaną skonstruowane większe układy CCD, umieszczenie kilku, powiedzmy czterech, układów jeden obok drugiego umożliwi skanowanie pasków o szerokości rzędu stopnia. I po roku czy dwóch powinniśmy mieć skompletowany przegląd CCD całego nieba. Jedyne problem to – kto będzie analizował tę super ogromną ilość danych. Paul odpowiada z uśmiechem: komputery i studenci!

Przykłady programów imperatywnych w języku BASIC mogą Czytelnicy odnaleźć w dziale μ miesięcznika *Problemy*. My zilustrujemy myślenie imperatywne programem w języku PASCAL na generowanie tak zwanych *trójek pitagorejskich*. Trójką pitagorejską nazywa się w matematyce trójkę liczb naturalnych a, b, c spełniających warunek z twierdzenia Pitagorasa: $a^2 + b^2 = c^2$. Można wykazać, że takich trójek jest nieskończenie wiele. Nasz program ma generować i wyświetlać wszystkie trójki, w których c nie przekracza danej liczby M .

Zastosowana w programie metoda polega na przeglądaniu wszystkich par b, c , w których $c \leq M$ i $b < c$, wyliczeniu wartości pierwiastka z liczby $c^2 - b^2$ i zaokrągleniu jej do liczby naturalnej przez odrzucenie części ułamkowej. Następnie sprawdzamy, czy otrzymane w ten sposób liczby: a , które obliczyliśmy, b oraz c spełniają warunek Pitagorasa. Jeśli tak, to otrzymaną trójkę wyświetlamy na ekranie. Po zbadaniu danej pary b, c przechodzimy do kolejnej według zasady, że dla ustalonego c badane są wszystkie b w porządku rosnącym od 1 do $c - 1$. Po wyczerpaniu zaś wszystkich b następuje przejście do kolejnego, większego c , aż do wyczerpania liczb od 1 do M .

```

program TrójkiPitagorejskie;
const
    M = 10;
var
    a, b, c : Integer;

begin
for c := 1 to M do
    for b := 1 to c - 1 do

        begin
            a := trunc(sqrt(c*c - b*b));

            if a*a + b*b = c*c then
                writeln (a, b, c)
            end
        end
    end

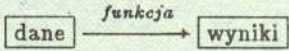
```

Maksymalna wartość zmiennej c.

Zmienne a, b, c mogą przyjmować liczbowe wartości całkowite. Początek instrukcji programu. Dla kolejnych wartości c od 1 do M wykonaj: przy ustalonym c dla kolejnych b od 1 do $c - 1$ wykonaj instrukcje pomiędzy słowami **begin** i **end**: przypisz zmiennej a wartość $\sqrt{c^2 - b^2}$ po odrzuceniu części ułamkowej, jeżeli otrzymana trójka jest pitagorejska – wyświetl liczby a, b, c .

Myślenie w kategoriach zbiorów i funkcji

W podejściu funkcyjnym program komputerowy jest traktowany jako funkcja, która w sposób jednoznaczny dla ustalonych danych wyznacza wyniki.



Programista ma możliwość myśleć w kategoriach pojęć matematycznych, takich jak zbiory i funkcje na zbiorach.