**13. Məktəb informatikasında struktur proqramlaşdırma üsulunun tədrisi metodikası**

Kompüterlə hər-hansı bir işin görülməsi üçün tələb olunan instruksiyalar ardıcıllığını tərtib etmək isə proqramlaşdırma adlanır.

İstənilən proqramlaşdırma dilinin əsas elementləri bunlardır: dilin əlifbası, sintaksisi və semantikası. Dilin əlifbası dedikdə, həmin dildə işlənən bütün simvollar nəzərdə tutulur. Sintaksis- əlifbada olan simvollardan dilin ayrı-ayrı konstruksiyalarının (komandaların, operatorların) düzəldilməsinin formal qaydalarıdır. Bu qaydalar müxtəlif həll alqoritmlərini proqramlaşdırmağa imkan verir.

**Strukturlu proqramlaşdırma**  — proqramların “ayır və buyur” ideyasına uyğun gələn layihələndirilməsi metodologiyasıdır. E.Deykstra (Edsger Dijkstra) tərəfindən işlənib hazırlanıb. Proqram məsələlər toplusu kimi layihələndirilir. Sadə təsvir üçün çox çətin olan məsələlər bir neçə daha kiçik altməsələlərə bölünür və proses bütün məsələlər yetərincə anlaşıqlı olanadək davam etdirilir.

Belə layihələndirmə nəticəsində proqram iyerarxik təşkil olunmuş və hər birinin yalnız bir giriş və bir çıxış nöqtəsi olan yığcam modullardan ibarət olacaq. Bundan başqa, strukturlu proqramlaşdırma GOTO deyimlərindən istifadəyə məhdudiyyət qoyur, çünki hesab edilir ki, proqramda xətaların sayı keçidlərin sayının kvadratına mütənasib artır. İri layihələrdə strukturlaşmış proqramları işləyib hazırlamaq və müşayət etmək asandır. Ancaq bu metodun bir sıra çatışmazlıqları var, məsələn, verilənlərin strukturları, onları idarə edən funksiyalardan ayrıldığından, verilənlər və onlar üzərində əməllər vahid tam kimi qavranılmır.

1971-ci ildə Nikuls Virt tərəfindən Paskal dili yaradılır. Bu dil struktur proqramlaşdırma ideyasının, yəni proqramın kiçik, dəqiq təyin edilmiş prosedurlardan tədricən qurulması ideyasının həyata keçirilməsini təmin edən ilk dildir.

Məsələn, Visual Basic-də Sətirlər və ya kod blokları ardıcıllıqla yazılır və yerinə yetirilir. Məsələn:

x = 6

y = 12

z = x + y

WriteLine (z)

**Təkrar**

Bir şərt yerinə yetirilərkən bir kod bloku təkrarlanır. Blokun icra olunma sayında heç bir məhdudiyyət yoxdur. Misal:

x = 2

X <100 olduğu müddətdə

WriteLine (x)

x = x \* x

Son

**Nümunə.** Aşağıdakı kod 1-dən 100-ə qədər ədəd arasında istifadəçinin daxil etdiyi ədədə bölünən ədədləri və onların sayını çap edir.

#include <iostream>

int main **(){**

int k**,** x**,** say**;**

cout **<<** "Her-hansi eded daxil edin \n"**;** cin **>>** x**;**

cout **<<** "1 ile 100 arasinda "**<<** x

1. " -e bolunen ededler:\n\n"**;** say **=** 0**; for (** k**=**1**;** k**<**100**;** k**++ )**

**if (**k **%** x **==** 0**) {**

cout **<<** k **<<** " "**;**

say**++;**

**}**

cout **<<** "\n Cemi " **<<** say **<<** " eded \n"**;**

**}**

**Nəticə:**

Her-hansi eded daxil edin**:** 23

1. ile 100 arasinda 23 -e bolunen ededler**:**

23 46 69 92

Cemi 4 eded