

MATEMATİKSEL BİR PROGRAMLAMA DİLİ: ISETL

Serpil KONYALIOĞLU*

Özet

Bu çalışmada, matematiksel bir programlama dili olan ISETL programlama dili hakkında kısaca bilgi verilmiştir.

Anahtar Kelimeler: ISETL, Matematiksel Programlama Dili

A MATHEMATICAL PROGRAMMING LANGUAGE: ISETL

Abstract

In this study, it was briefly informed about ISETL programming language, a mathematical programming language,

Key Words: ISETL, Mathematical Programming Language

Giriş

ISETL, 1969 yılında bilgisayar bilimcisi ve aynı zamanda bir matematikçi olan Jack Schwartz tarafından geliştirilen SETL programının daha ileri bir versiyonudur. Orijinal adı *Interactive SET Language* yani *Etkileşimli Küme Dili* olan bu programlama dili, 1985 yılında Gary Levin tarafından geliştirilmiştir. Gary Levin'in Unix, DOS ve Macintosh işletim sistemlerinde kullanılabilecek şekilde geliştirdiği bu programlama dilinin Windows versiyonu ise 1995 yılında John F. Kirchmeyer tarafından geliştirilmiştir.

1985 yılına kadar SETL programı olarak sadece yüksek kapasiteli ve pahalı bilgisayarlarda çalışabilen bu dil, yeni sürümlerin ortaya çıkması ve kişisel bilgisayarların yaygınlaşmasıyla tüm bilgisayarlarda çalıştırılabilir bir

* Atatürk Üniversitesi, K.K.Eğitim Fakültesi, Ortaöğretim Matematik Eğitimi Anabilim Dalı ,
Doktora Öğrencisi

program haline getirilip ISETL adı verilmiştir. SETL programının interaktif versiyonu olan ISETL, daha hızlı, hatasız çalışan ve temel objeleri fonksiyon ve prosedürler olan matematiksel bir programlama dilidir.

Bir programlama dilinin matematiksel bir programlama dili olabilmesi için bazı özelliklere sahip olması gerekir. Özellikle kod ve temel kalıplar standart matematiksel notasyonlarla benzer olmalıdır. Matematik için gerekli olmayan söz dizimleri ve kodlar programda olmamalı ya da nadiren kullanılmalıdır. Programlama dilinde kullanılan matematik öğeleri, matematiksel özelliklerle uyum içinde olmalıdır. Ayrıca programdaki temel objeler, matematiği öğrenmeye yardımcı olacak ifadelerde kullanılan öğeler olmalıdır. Bu özellikleri az ya da çok sağlayan bir programlama dilinin matematiksel bir programlama dili olduğu söylenebilir. ISETL bu özelliklere sahip bir programlama dilidir (Dubinsky 1995).

ISETL, matematikteki küme ve fonksiyon kavramlarına dayanan matematiksel bir programlama dilidir. Bu programlama dili; tamsayılar, ondalık sayılar, boolean gibi basit data tipleri yanında küme, sıralı n-li ve func gibi bileşik data tiplerini de desteklemektedir.

ISETL programlama dilindeki sintaks ve temel kodlar, matematik notasyonları ve kavramları ile uyumluluk içindedir. Örneğin, ISETL programlama dilindeki temel objelerden biri olan küme, matematikte olduğu gibi elemanların bir topluluğudur ve yine matematikte olduğu gibi ISETL programlama dilinde de küme tanımlamanın bir yolu; elemanların küme parantezinde yazılmasıdır. Küçük kümeler için tercih edilen bu yolun yanında daha büyük kümeleri yazmak için;

Küme adı := { ifade: elemanların yeri: koşul(isteğe bağlı)}; şablonu kullanılmaktadır. Bu şablondaki ifade, kümedeki elemanların nelerden oluştuğunu; elemanların yeri, kümedeki elemanların nereden alındığını; koşul ise isteğe bağlı olarak yazılan ve elemanların sahip olduğu ek bir özelliği göstermektedir. Örneğin;

> A := {x: x in {0..50}: x mod 2 = 0};

kümesi, 0 ile 50 arasındaki çift tamsayılardan oluşmaktadır. Bu kümenin elemanlarını listelemek için;

```
> A;
```

yazmak gerekmektedir.

Sıralı n-li objesi de, matematikteki n-bileşenli kavramı ile aynıdır. Sıralı n-liler de kümelerle aynı şekilde tanımlanmaktadır. Bu objenin kümeden farkı, elemanların sırasının önemli olmasıdır. Bu öge, ISETL programlama dilinde küme parantezi yerine köşeli parantez kullanılarak yazılmaktadır. Örneğin;

```
> B := [n**3: n in [1..100]];
```

sıralı kümesi, 1 ile 100 arasındaki sayıların küplerinden oluşmaktadır.

ISETL programlama dilinde öğrencilerin doğru programlama yapabilmeleri için, öğrenmeleri gereken bazı kodlar ve kalıplar bulunmaktadır. Bu kalıplar diğer programlama dillerindeki kalıplarla benzer nitelikte olup öğrencilerin kelime hazinesiyle tutarlı anlamlara sahiptir. Örneğin bir G kümesinin 'o' işlemine göre kapalı olup olmadığını araştıran bir öğrenci;

```
forall a, b in G: a.ob in G;
```

şeklinde bir program yazmalıdır. Öğrencinin bu programlamayı yaparken yazılıma ait bilmesi gereken tek kod forall niceleyicisidir. forall yani "her" niceleyicisi, öğrencinin mevcut bilgisi ile çelişmeyen ve kapalılığın tanımında bulunan bir ifadedir. Yazılımın kodlarını ve sintakslarını öğrenen bir öğrenci, aslında kavramları oluşturan temel yapı taşlarını da öğrenmiş olmaktadır.

ISETL programlama dilinde yazılan programlar standart matematik diline çok benzediği için, öğrenci ISETL programlama dilinin kullanımını öğrendiğinde matematiksel ifadeleri yazmayı da öğrenmektedir. Üstelik matematiksel bir kavramı bilgisayarda oluşturmaya çalışmak kavramı öğrenmek için mükemmel bir yoldur (Fenton and Dubinsky 1996). Örneğin,

$$\forall a \in G \exists b \in G \text{ öylki } a + b = 0$$

ifadesi ISETL programlama dilinde;

forall a in G: exists b in G: a + b = 0;

ifadesine karşılık gelir. Yine,

$\{x \in \{0,1,\dots,100\} | x^2 \text{ dört ile bölünebilir} \}$

kümesi ISETL’de;

$\{x: x \text{ in } \{0..100\}: x^{**2} \text{ mod } 4 = 0\}$

şeklinde yazılabilir.

ISETL dilinde, ileri derecedeki bir çok matematiksel kavramın programlanmasında fonksiyonlardan yararlanır. Bu nedenle fonksiyonlar, ISETL programlama dilinde birinci sınıf objedir. ISETL programlama dilinde, bir fonksiyon genel olarak dört kısımdan oluşur: fonksiyonunu adı, girdi değişkenlerinin listesi, çıktının nasıl olması gerektiğini gösteren emirler ve end ifadesi. Sintaks olarak func;

```
isim := func(girdi değişkenlerinin listesi);
      return ifadesi ve bazı emirler
end;
```

şeklinde yazılabilir. Örneğin, sayı doğrusunda iki nokta arasındaki uzaklığı hesaplayan bir fonksiyon;

```
> F := func(x,y);
>> if x > y then return x-y;
>> else return y-x;
>> end;
>> end;
```

şeklinde programlanabilir.

func, ISETL için temel objedir. Yani bir çok kavramın yazımında, kavramların özelliklerini test etmede ve örnek çözümünde bu objeden yararlanır. Örneğin, bir G kümesi üzerinde tanımlanan ‘o’ işleminin değişmeli olup olmadığını belirlemek için func kullanılarak şu kod yazılmalıdır:

```
değişme := func(G,o);
return
```

```
forall a,b in G | a .o b = b .o a;  
end;
```

Bu kısa program yazıldıktan sonra;

>değişme (G,o):

kodu ile G kümesinin 'o' işlemine göre değişmeli olup olmadığı belirlenebilir. G kümesi 'o' işlemine göre değişmeli ise cevap 'true', değişmeli değilse cevap 'false' olacaktır. 'o' işleminin sahip olduğu ya da olmadığı diğer tüm özellikler, yukarıdaki gibi kısa programlar yazılarak tespit edilebilir.

ISETL programlama dilinin temel sintaksları matematiğin diline benzer olduğundan, öğrencilerin bu programlama dilini öğrenmesi de kolay olmaktadır. Dubinsky ve Yiparaki (1996), ISETL'nin öğretimi için, çok kısa bir zaman diliminin yeterli olabileceğini, çünkü öğrencilerin ISETL'de programlama yapabilme yeteneğinin, matematiksel oluşumları bilmelerinden kaynaklandığını belirtmişlerdir.

Leron ve Dubinsky (1995), matematik öğrenmek için, daima matematiksel prensiplerin, objelerin ve bunlar arasındaki ilişkilerin kurulması yani oluşturulması gerektiğini ve bu oluşumlar için temel öğelerin de, genellikle, kümeler, fonksiyonlar, niceleyicilerle önceki oluşumların sonuçları yani kavramla ilgili önkoşul kavramlar olduğunu belirtmişler ve ISETL programlama dilinin de, bu tür oluşumları ifade edebilmek için tasarlanmış özel bir programlama dili olduğunu ifade etmişlerdir.

Öğrenciler yazılımla ilk tanıştıklarında tam sayılar, ondalık sayılar ve Boolean değerlerinde işlem yapmak için kodlar yazmalıdır. Bu da matematiksel objeler üzerinde işlem yapma becerisini geliştirmektedir (Dubinsky 1995).

Öğrenciler yazılımla biraz zaman geçirdiklerinde, küme, sıralı ikili ve sıralı n-li kodlarını öğrenirken sayılar ve sayılarla yapılan işlemleri öğrenmenin daha ileri bir aşamasına geçerek küme ve sıralı n-li kavramını özümseyebilmektedirler. Ayrıca ISETL'de func öğesini programlamayı öğrenmek, öğreneilerin fonksiyon örnekleri repertuarını

geniřtebilmektedir. Bylelikle ISETL ortamında yapılan etkinliklerin eřidi de artmaktadır.

ISETL programlama dilinde, yapılan programlamanın yanlıřlıđı veya eksikliđi hakkında hemen dnt alınabilmektedir. Verilen dntler dođrultusunda hatalar dzeltilebilmektedir. đrencilerin bilgilerini inřa sürecinde yaptıkları hatalarla ilgili geri bildirim almaları, onların yeni dřnme yolları aramasına neden olmaktadır.

ISETL programlama dili kullanıcıya hazır kullanım imkanı veren bir yazılımdır. đrenciler yazılımı bilgisayara ykledikten sonra minimum dzeyde iřletim sistemi becerilerini kullanmak durumundadırlar. Kullanıcının ileri dzeyde bilgisayar iřletim sistemi bilmesini gerektiren yazılımlar etkili yazılımlar olmadıkları iin, ISETL đrencinin mdahalesini minimuma indirecek řekilde tasarlanmıřtır. Ayrıca, ISETL programlama dili bilgisayarda ok fazla yer kaplamayan ve bir diskete sığabilme zelliđi ile de tercih edilebilir bir programlama dilidir.

ISETL programlama dili ile uygulama yapan đrenciler dersin sonunda yaptıkları tm programları ve uygulamaları kendi adlarına atırları dosyalara kaydedebilmektedirler. Ayrıca bir đretmenin đretim ortamında gsterdiđi birok faaliyeti, yani đrenciye alıřtırma ve tekrar yaptırma, geri bildirim sađlama gibi aktiviteleri ISETL programlama dili yapabilmektedir.

Sonuç

Matematik đretiminde pedagojik bir ara olarak tasarlanmış pek ok yazılım bulunmasına rađmen bu yazılımların ođu, kullanım sürecinde teknik ya da pedagojik problemlere neden olmaktadır. ISETL, tm bu eksiklik ve problemler gz nne alınarak geliřtirilmiş yeniliki bir programlama dilidir. niversite dzeyinde giriř niteliđindeki matematik derslerinin yanı sıra ileri ve soyut matematik derslerinde de kullanılan bu yazılım dayandıđı pedagojide gz nne alınarak mfredata eklendiđinde, eđitim-đretim sürecini zenginleřtirebilecek nemli bir teknolojik aratır.

Kaynaklar

- Dubinsky, E., 1995. ISETL: A programming language for learning mathematics. *Communications on Pure and Applied Mathematics*, 48. 1-25.
- Dubinsky, E., Yiparaki, O., 1996. Predicate calculus and mathematical thinking of students, DIMACS Symposium: Teaching Logic and Reasoning in Illogical World, 25-26 July, Rutgers University, Piscataway, New Jersey.
- Leron, U. and Dubinsky, E., 1995. An abstract algebra story. *American Mathematical Monthly*, 102 (3). 227-242.
- Fenton, W. E. and Dubinsky, E., 1996. Introduction to discrete mathematics with ISETL. Springer-Verlag, New York..