

Ahmet Emin Kaplan – 131044042 – CSE 222 HW04 REPORT

Öncelikle ödevi yapmaya başlamadan önce dizayn konusunda expressionlari daha rahat opere edebilmek için Operand ve Operator sınıfları oluşturdum .Oluşturmuş olduğum bu sınıflar Expression isimli abstract bir sınıftan extend olmakta.Ödevde inputlar dosyadan alındığı için , satır satır okuma yaptım ve okuduğum her bir satırı kendi algoritmamla parse ettim yani operatör ve operand içeren stringi parçaladım ve bu verileri bir arraylistte tutarak Temel sınıfım olan ExpressionConverter objeme gönderdim.Temel sınıfım olan ExpressionConverter bazı metodlara sahip bu metodlar ın başında infix ifadeyi postfix forma çeviren InfixToPostfix() metodu bulunmakta .Infix ifadeyi postfix forma çevirmek için stack yapısını kullandım ve operatörlerin öncelik sırasına göre postfix ifademide elde edebildim.Elde ettiğim postfix ifade oluşturmam gereken assembly ifadesini elde etmemde bana kolaylık sağlayacak çünkü postfix ifadem işlem önceliklerinden münezzehe bir durumda.ExpressionConverter sınıfı kendi içinde has a relation ile Registerlara sahip .Postfix olarak elde ettiğim expression ı assembly koduna çevirirken dikkat ettiğim öncelikli husus sahip olduğum registerlari idareli kullanmak. Assembly kodu üretirken gerekli +-*/ ve = operatörlerinin karşılık geldiği ifadeleri PDF de belirtilen durumlara göre ayarladım.Gerekli durumlarda error handling mekanizmalarini kullandım örneğin Divide by zero , tanımlanmayan bir değişkenin operand olarak kullanılması, constant bir değere atama yapmak,bütün registerların dolu olması gibi durumlarda Exception fırlatıyorum.Yukarda belirttiğim operatörlerin her biri için özel metodlar yazdım bu metodlar işlemi yaparken kullanılacak operandların kontrolünü işlem esnasında registerların durumunu işlem sonucunun kayıtlanması gibi durumları opere etmekte.print komutu için özel bir metod yazmadım print komutunu InfixToPostfix() metodu içinde handle ettim .Bu object oriented metodolojiye biraz aykırı olabilir ama başlangıçta kurduğum dizaynın çok güçlü olmaması beni bu duruma itti.Assembly koduna çevirdiğim her bir process ExpressionConverter sınıfımın sahip olduğu bir stringe(FILEOUTPUT) eklenerek saklanmakta.Eğer herhangi bir exception oluşmazsa bütün operasyonlar yapıldıktan sonra Print() metodu çağrılarak output.asm dosyasına elde edilen instructionlar yazılmakta.Yazdığım kodu çeşitli inputlarla denedim herhangi bir problemle karşılaşmadım ayrıca yazdığım kodu UnitTestlerle kontrol ettim.Javadoc dokümantasyonu ile metodlarımın özelliklerini belirttim.Bu ödevde süre konusunda sıkıntılar yaşadım ama **Erdoğan hocanın bize verdiği 1 kereye mahsus 2gün gecikmeli (cezasız) submission hakkını** kullandığım için ucu ucuna yetiştirmiş bulunuyorum.

Ödev esnasında Assembly hakkında yüzeysel bilgilerle kendimi geliştirdiğimi düşünüyorum.Ayrıca Postfix expression ve Stack yapısı hakkında uygulamalı pratik bilgiler edindim.