

Ödev 1

Son teslim tarihi: 17 Kasım 2024

Proje Tanımı: ANSI Terminal Tabanlı Elektronik Tablo Programı

Genel Bakış

Bu projede, VisiCalc'tan esinlenerek basitleştirilmiş bir elektronik tablo programı oluşturacaksınız. Program bir ANSI terminalinde çalışacak ve metni görüntülemek ve kullanıcı girdisini almak için sağlanan AnsiTerminal sınıfını kullanacaktır. C++ temellerini, OOP kavramlarını, STL kullanımını ve dosya işlemeyi anladığınızı göstereceksiniz. Amaç, temel hesaplamaları yapabilen, hücre referansını destekleyen ve CSV formatında veri kaydedip yükleyebilen işlevsel bir elektronik tablo uygulaması oluşturmaktır. Program, hücre referanslama, formüller ve otomatik yeniden hesaplama gibi temel hesap tablosu özelliklerini içerecektir.

Gereksinimler

Proje aşağıdaki gereklilikleri içerecek olup, öğrenciler bu gereklilikleri yerine getirmeli ve teslimlerinin bir parçası olarak bir PDF dosyasında belgelemelidir.

1. Dokümantasyon (PDF Dosyası)

- UML Sınıf Diyagramı: Elektronik tablo uygulamasının sınıf yapısını temsil eden ve anahtar sınıflar arasındaki ilişkileri gösteren bir UML diyagramı oluşturun ve ekleyin. UML diyagramları üretmek için plantuml.com adresini kullanın.
- Uygulanan Özellikler: Uygulanan özellikleri açıklayın ve eksik özellikleri açıkça listeleyin.
- Yapay Zekâ Tabanlı Yardım: ChatGPT veya diğer yapay zekâ tabanlı araçlardan yardım alınan kod bölümlerini not edin.
- Kullanıcı Kılavuzu: Örnek komutlar ve beklenen sonuçlarla birlikte programın nasıl kullanılacağını açıklayan bir kullanıcı kılavuzu ekleyin.

2. Temel Hesaplamalar ve Formüller

- Kullanıcılar sayıları, stringleri girebilmeli ve hücrelerde temel aritmetik hesaplamaları yapabilmelidir (örneğin, $=A1 + B2 - C3$).
- Program, formüller içinde toplama, çıkarma, çarpma ve bölme işlemlerini desteklemelidir. Parantezlere izin verilmez.

3. Otomatik Yeniden Hesaplama

- Bir hücre değeri değiştirildiğinde elektronik tablo bağlı hücreleri otomatik olarak güncellemelidir. Döngüsel referans sorununu göz ardı edin.
- Örneğin, B2 hücresi =A1 + 5 içeriyorsa, A1 değerinin değiştirilmesi B2 değerini otomatik olarak yeniden hesaplamalıdır.

4. Sütun ve Satır Etiketleri

- Veri takibini kolaylaştırmak için satırlar sayısal olarak (1, 2, 3, ...) ve sütunlar alfabetik olarak (A, B, C, ..., AA, AB, ...) tipik elektronik tablo yazılımına benzer şekilde etiketlenecektir.

5. Hücre Referanslama

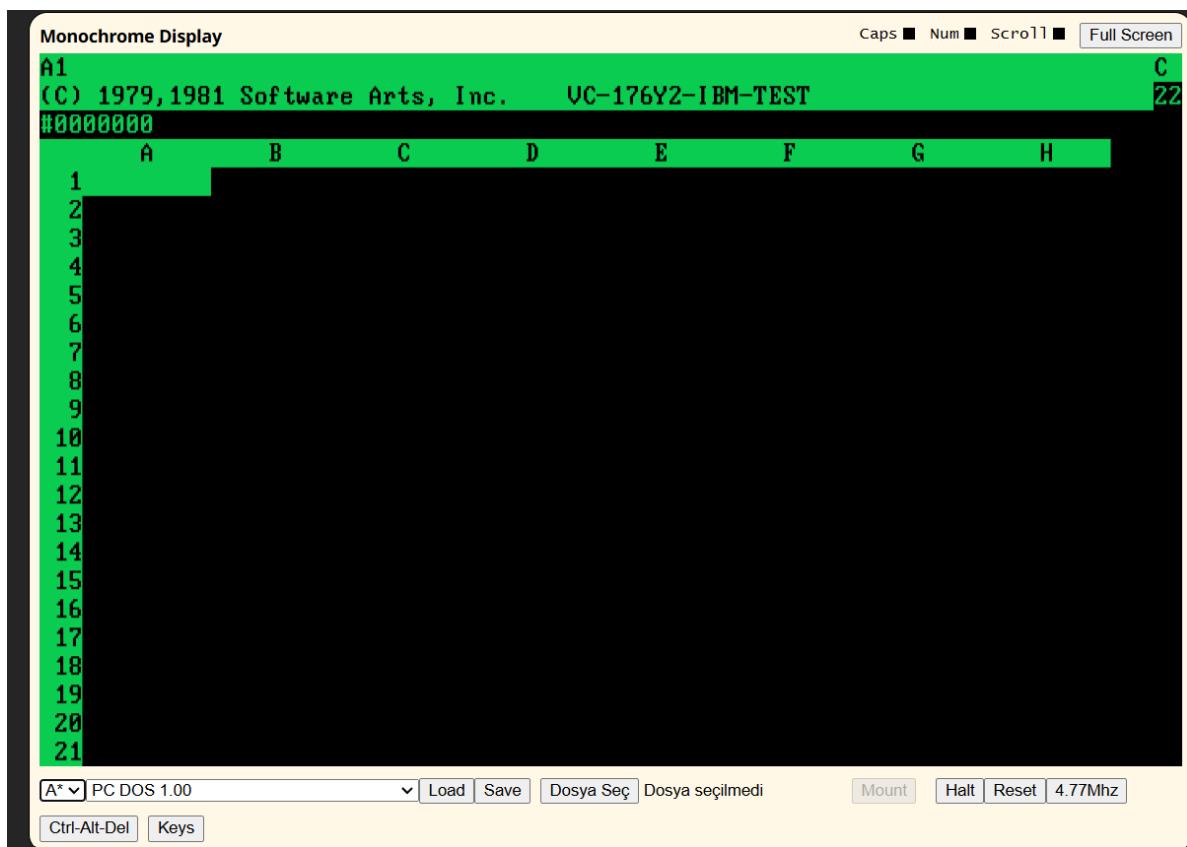
- Kullanıcılar, hücre tanımlayıcılarını (örneğin, B1 + C1) kullanarak formüller içinde diğer hücrelere başvurabilir.
- Program mutlak hücre referansını destekler ancak görelî hücre adreslemesini desteklemez.
- Desteklenen fonksiyonlar:
 - TOPLA: Bir hücre aralığını toplar, örneğin, TOPLA(A1..A10).
 - AVER: Bir hücre aralığının ortalamasını alır, örneğin AVER(A1..A10).
 - STDDEV: Bir hücre aralığının standart sapmasını hesaplar, örneğin, STDDEV(A1..A10).
 - MAX: Bir aralıktaki maksimum değeri verir, örneğin, MAX(A1..A10).
 - MIN: Bir aralıktaki minimum değeri verir, örneğin, MIN(A1..A10).
- Örnek formül: B1 - MIN(A1..A10). Formüllerde parantez kullanılmayacaktır.

6. Görsel Arayüz

- Kullanıcı arayüzü VisiCalc'ın şu adreste gösterilen düzenebine benzemelidir:

<https://www.pcjs.org/software/pcx86/app/other/visicalc/1981/>

- En üst satır, seçili hücrenin içeriğini görüntüler ve aynı zamanda kullanıcıların komutları ve formülleri yazdıkları giriş alanı olarak işlev görür.



7. Dosya İşlemleri (CSV Formatı)

- Elektronik tablo verilerini CSV formatında kaydetmek ve yüklemek için fonksiyonellik uygulayın. Bu, Microsoft Excel gibi diğer elektronik tablo uygulamalarıyla uyumluluk sağlar.
- Elektronik tablonun her satırını .csv dosyasında virgülle ayrılmış değerlerden oluşan bir satır olarak kaydedin.
- Unicode karakterlerin desteklenmesi gerekmek.
- Yeni dosya oluşturma, dosya kaydetme, dosya yükleme, dosyayı farklı kaydetme gibi dosya işlemleri uygulanmalıdır.

8. Veri Depolama

- Elektronik tablo ızgarası için hücre verilerini depolamak üzere bir 2D std::vector kullanın. Vektördeki her hücre bir sayı, formül veya string değeri içerecektir.

Teslim Koşulları

- Kaynak Kodu: Sağlanan AnsiTerminal.h ve AnsiTerminal.cpp dosyaları da dahil olmak üzere tüm kaynak dosyalar.
- Her sınıf için bir başlık dosyası ve bir CPP dosyası ekleyin.
- Dokümantasyon PDF:
 - Sınıf yapısının UML Diyagramı.
 - Uygulanan özelliklerin ve eksik parçaların açıklaması.
 - Varsa yapay zekâ yardımı beyanı.
 - Programın kullanımını örneklerle açıklayan bir Kullanım Kılavuzu.
- Standart C kütüphanelerinden herhangi bir fonksiyon kullanmayın (printf gibi), C arrayleri kullanmayın. Matematik fonksiyonları için standart C fonksiyonlarını kullanabilirsiniz.
- String, vector, list gibi C++ standart sınıflarını kullanın.
- Derslerde öğrendiğimiz consts, C++11 özellikleri gibi tüm OOP tekniklerini kullanın (range for döngüleri, strong enums, auto anahtar sözcüğü, decltype anahtar sözcüğü, vb.)
- Kodunuzu girintilemeyi (düzen) ve anlamlı yorumlar yapmayı unutmayın.
- Programınızı test etmek için bir dizi CSV dosyası sağlayacağınız
- Çalışmanızı, asistanların talimatlarını kullanarak Teams sayfasına yüklemelisiniz.
- Ödevlerinizi çevrimiçi olarak demo yapacaksınız

Önerilen Zaman Çizelgesi

1. Hafta: Proje kurulumu, class diyagramı ve ilk Spreadsheet ve Cell sınıfının geliştirilmesi.
2. Hafta: FormulaParser sınıfı ile formül ayrıştırma ve hücre referansı uygulama.
3. Hafta: CSV için FileManager sınıfının geliştirilmesi: CSV işlemleri için FileManager sınıfının geliştirilmesi ve kalan özelliklerin tamamlanması.

Ayrıca test, hata düzeltme ve son dokümantasyonu yapın.