2024-2025 GÜZ ÖĞRETİM DÖNEMİ



GAZİ ÜNİVERSİTESİ TEKNOLOJİ

FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BMT- 103 KODLU DERSİN PROJE RAPORU

ŞUBE : 1 GRUP NO : 4

PROJE ADI : Bir sigorta şirketindeki çalışanların aylık satış değerinin satış performans değerine çevrilmiş halini gösteren program

HAZIRLAYANLAR :

|  |  |
| --- | --- |
| Öğrenci No | İsim - Soyisim |
| 23181616028 | Fatma SANSA |
| 23181616058 | Hazalnur DAČLI |
| 23181617014 | Atacan PINAR |

1

Proje Konusu => Çalışan sayısı 12 olan bir sigorta şirketinde , çalışanlarının soyadları , ad ve aylık satış bilgileri satis.txt adlı bir metin dosyasının içinde tutulmaktadır . Bu dosyanın içinde bulunan bilgilerin

okunarak prfrmns.txt adlı dosyanın içerisine çalışanlarının bilgilerini ve aylık satış değeri yerine satış performans

değerine çevrilmiş halini kaydeden programın yazılması.

Proje Amacı => Çalışanların aylık satış bilgilerini

kullanarak performans değerini hesaplamak . Şirketin işine yarayabilecek bilgileri bir arada tutarak hem de alfabetik düzeni sağlayarak karışıklığın önüne geçerek istenen değerlerin elde edilmesi.

Proje Toplantısı =>

30.11.2024 tarihinde görüşüldü projenin genel durumu tartışıldı .

05.12.2024 tarihinde buluşulup görev dağılımı yapıldı.

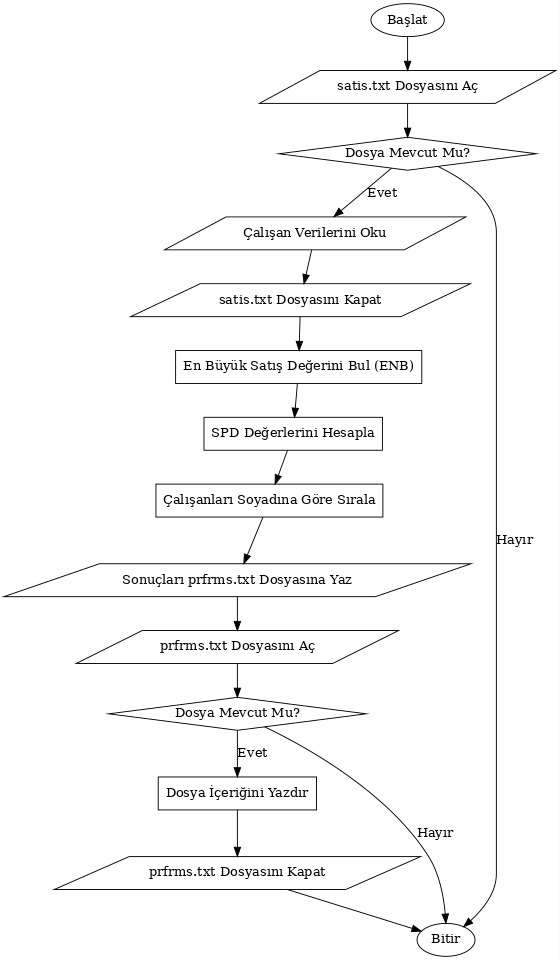
12.12.2024 tarihinde yapılan işlemler ekipce kontrol

Edildi.

20.12.2024 tarihinde son kontroller yapıldı.

2

AKIŞ DİYAGRAMI :



3

KODLAR :

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

// Çalışan bilgilerini tutan struct

typedef struct {

char soyad[50]; // Çalışanın soyadı

char ad[50]; // Çalışanın adı

double satis; // Çalışanın aylık satış değeri

double spd; // Çalışanın satış performans değeri

} personel;

// Fonksiyonlar

double enBuyukSatis(personel p[], int n); // En büyük satış değerini bulur

void alfabetikSirala(personel p[], int n); // Çalışanları soyadına göre sıralar

void dosyayaYaz(personel p[], int n, const char \*dosyaAdi); // Çalışan bilgilerini bir dosyaya yazar

int main() {

FILE \*dosya; // Dosya işaretçisi tanımlanır

personel calisanlar[12]; // Maksimum 12 çalışan bilgisi tutacak dizi

int i = 0, n = 0; // Sayaç değişkenleri ve çalışan sayısı

double ENB; // En büyük satış değeri için değişken

const char \*dosyaYolu = "C:\\Users\\ataca\\OneDrive\\satis.txt"; // Giriş dosyasının yolu

const char \*prfrmsDosyaYolu = "C:\\Users\\ataca\\OneDrive\\prfrms.txt"; // Çıkış dosyasının yolu

// satis.txt dosyasını aç

dosya = fopen(dosyaYolu, "r");

if (dosya == NULL) {

// Dosya açılmazsa hata mesajı yazdırılır

printf("Hata: satis.txt dosyasi bulunamadi! Lutfen dosyayi manuel olarak olusturup calistirin.\n");

return 1; // Program sonlandırılır

}

// satis.txt dosyasındaki veriyi oku

while (fscanf(dosya, "%s %s %lf", calisanlar[i].soyad, calisanlar[i].ad, &calisanlar[i].satis) == 3) {

i++; // Okunan çalışan sayısını artır

}

n = i; // Toplam çalışan sayısını kaydet

fclose(dosya); // Dosyayı kapat

// En büyük satış değerini bul

ENB = enBuyukSatis(calisanlar, n);

// Satış performans değerlerini hesapla

for (i = 0; i < n; i++) {

calisanlar[i].spd = ENB - calisanlar[i].satis; // Performans değerini hesapla

}

// Çalışanları soyadlarına göre alfabetik sırala

alfabetikSirala(calisanlar, n);

// Sonuçları prfrms.txt dosyasına yaz

dosyayaYaz(calisanlar, n, prfrmsDosyaYolu);

// prfrms.txt dosyasını ekrana bastır

dosya = fopen(prfrmsDosyaYolu, "r");

if (dosya == NULL) {

// Çıkış dosyası açılamazsa hata mesajı yazdır

printf("Hata: prfrms.txt dosyasi acilamadi!\n");

return 1; // Program sonlandırılır

}

// Dosyanın içeriğini ekrana yazdır

printf("\nprfrms.txt dosyasinin icerigi:\n");

char satir[100]; // Satır okuma tamponu

while (fgets(satir, sizeof(satir), dosya) != NULL) {

printf("%s", satir); // Her bir satırı ekrana yazdır

}

fclose(dosya); // Dosyayı kapat

printf("\nIslem tamamlandi. Sonuclar yukarida ekrana bastirildi ve prfrms.txt dosyasina yazildi.\n");

return 0; // Program başarıyla sonlandı

}

// En büyük satış değerini bulan fonksiyon

double enBuyukSatis(personel p[], int n) {

double max = p[0].satis; // İlk çalışanın satışını başlangıç değeri olarak al

for (int i = 1; i < n; i++) {

if (p[i].satis > max) { // Daha büyük bir satış değeri bulunursa

max = p[i].satis; // Maksimum değeri güncelle

}

}

return max; // En büyük satış değerini döndür

}

// Soyada göre alfabetik sıralama yapan fonksiyon (Bubble Sort kullanıldı)

void alfabetikSirala(personel p[], int n) {

personel temp; // Geçici değişken

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (strcmp(p[j].soyad, p[j + 1].soyad) > 0) { // Alfabetik sırayı kontrol et

temp = p[j];

p[j] = p[j + 1];

p[j + 1] = temp; // Yer değiştir

}

}

}}

// Sonuçları dosyaya yazan fonksiyon

void dosyayaYaz(personel p[], int n, const char \*dosyaAdi) {

FILE \*dosya = fopen(dosyaAdi, "w"); // Dosya yazma modunda aç

if (dosya == NULL) {

// Dosya açılamazsa hata mesajı yazdır

printf("Dosya yazma hatasi!\n");

return;

}

for (int i = 0; i < n; i++) {

// Her bir çalışanın bilgilerini dosyaya yaz

fprintf(dosya, "%s %s %.2lf\n", p[i].soyad, p[i].ad, p[i].spd);

}

fclose(dosya); // Dosyayı kapat

printf("Sonuclar %s dosyasina yazildi.\n", dosyaAdi); // Bilgilendirme mesajı

}

ÇIKTILAR:

