

```

class Market:
    def __init__(self):
        self.file_name = "product.txt"
        # Dosya mevcut değilse oluştur ve örnek ürünleri ekle
        try:
            with open(self.file_name, 'r') as file:
                pass
        except FileNotFoundError:
            with open(self.file_name, 'w') as file:
                products = [
                    "Pantolon,Kıyafet,120,50",
                    "Pantolon,Kıyafet,150,30",
                    "Pantolon,Kıyafet,200,20",
                    "Tişört,Kıyafet,50,100",
                    "Tişört,Kıyafet,60,80",
                    "Tişört,Kıyafet,70,60",
                    "Sweatshirt,Kıyafet,90,30",
                    "Sweatshirt,Kıyafet,110,20",
                    "Sweatshirt,Kıyafet,130,10",
                    "Ceket,Kıyafet,300,25",
                    "Ceket,Kıyafet,350,20",
                    "Ceket,Kıyafet,400,15",
                    "Eşofman,Kıyafet,80,50",
                    "Eşofman,Kıyafet,90,40",
                    "Eşofman,Kıyafet,100,30",
                    "Kalem,Kırtasiye,10,100",
                    "Kalem,Kırtasiye,15,80",
                    "Kalem,Kırtasiye,20,60",
                    "Sırt Çantası,Kırtasiye,200,30",
                    "Sırt Çantası,Kırtasiye,250,20",
                    "Sırt Çantası,Kırtasiye,300,15",
                    "Silgi,Kırtasiye,5,200",
                    "Silgi,Kırtasiye,7,150",
                    "Silgi,Kırtasiye,10,100",
                    "Defter,Kırtasiye,20,100",
                    "Defter,Kırtasiye,25,80",
                    "Defter,Kırtasiye,30,60"
                ]
                file.write("\n".join(products))

    def __del__(self):
        print("Program sonlandırıldı. Dosya kapatıldı.")

    def list_products(self):
        try:
            with open(self.file_name, 'r') as file:
                lines = file.readlines()
                if not lines:
                    print("Ürün listesi boş.")
                    return

            print("\nSıralama Seçenekleri:")
            print("1) Alfabetik (A-Z)")
            print("2) Alfabetik (Z-A)")
            print("3) Fiyat (Artan)")
            print("4) Fiyat (Azalan)")
            print("5) Kategori")
            sort_choice = input("Seçiminizi yapın (1-5): ")

            if sort_choice == "1":

```

```

        sorted_lines = sorted(lines, key=lambda x: x.split(',')[0])
    elif sort_choice == "2":
        sorted_lines = sorted(lines, key=lambda x: x.split(',')[0], reverse=True)
    elif sort_choice == "3":
        sorted_lines = sorted(lines, key=lambda x: float(x.split(',')[2]))
    elif sort_choice == "4":
        sorted_lines = sorted(lines, key=lambda x: float(x.split(',')[2]), reverse=True)
    elif sort_choice == "5":
        sorted_lines = sorted(lines, key=lambda x: x.split(',')[1])
    else:
        print("Geçersiz seçim, varsayılan sıralama (A-Z) uygulanıyor.")
        sorted_lines = sorted(lines, key=lambda x: x.split(',')[0])

    print("\n--- Ürün Listesi ---")
    for i, line in enumerate(sorted_lines, start=1):
        parts = line.strip().split(',')
        if len(parts) == 4:
            name, category, price, stock = parts
            print(f"{i}. Ürün: {name}, Kategori: {category}, Fiyat: {price} TL, Stok: {stock}")
        else:
            print(f"{i}. Hatalı format: {line.strip()}")
except Exception as e:
    print("Bir hata oluştu:", e)

def add_product(self):
    try:
        name = input("Ürün adı: ")
        category = input("Kategori: ")

        while True:
            price = input("Fiyat: ")
            try:
                price = float(price)
                if price > 0:
                    break
            else:
                print("Fiyat pozitif bir sayı olmalıdır.")
        except ValueError:
            print("Lütfen geçerli bir sayı girin.")

        while True:
            stock = input("Stok miktarı: ")
            try:
                stock = int(stock)
                if stock >= 0:
                    break
            else:
                print("Stok miktarı negatif olamaz.")
        except ValueError:
            print("Lütfen geçerli bir tamsayı girin.")

        product_line = f"{name},{category},{price},{stock}\n"
        if len(product_line.strip().split(',')) == 4:
            with open(self.file_name, 'a') as file:
                file.write(product_line)
            print("Ürün başarıyla eklendi.")
        else:
            print("Ürün formatı hatalı, eklenemedi.")
    except Exception as e:
        print("Bir hata oluştu:", e)

```

```

def delete_product(self):
    try:
        self.list_products()
        product_indexes = input("Silmek istediğiniz ürünlerin numaralarını virgülle ayırarak girin
(örn: 1,3,5): ")
        indexes = sorted([int(idx.strip()) - 1 for idx in product_indexes.split(',') if idx.strip().isdigit()],
reverse=True)

        with open(self.file_name, 'r') as file:
            lines = file.readlines()

        for index in indexes:
            if 0 <= index < len(lines):
                deleted_product = lines.pop(index)
                print(f"{deleted_product.strip()} ürünü silindi.")
            else:
                print(f"Geçersiz ürün numarası: {index + 1}")

        with open(self.file_name, 'w') as file:
            file.writelines(lines)
    except ValueError:
        print("Lütfen geçerli bir sayı girin.")
    except Exception as e:
        print("Bir hata oluştu:", e)

# Menü
if __name__ == "__main__":
    market = Market()
    while True:
        print("\n*** MENÜ ***")
        print("1) Ürünleri Listele")
        print("2) Ürün Ekle")
        print("3) Ürün Sil")
        print("4) Çıkış")

        choice = input("Seçiminizi yapın (1-4): ")

        if choice == "1":
            market.list_products()
        elif choice == "2":
            market.add_product()
        elif choice == "3":
            market.delete_product()
        elif choice == "4":
            print("Çıkış yapılıyor...")
            del market
            break
        else:
            print("Geçersiz seçim. Lütfen tekrar deneyin.")

```