



AKADEMIA GÓRNICZO-HUTNICZA
im. Stanisława Staszica w Krakowie



WYDZIAŁ ZARZĄDZANIA

Amelia Madej, Weronika Duda, Natalia Kargól
Projekt zaliczeniowy z “Programowania Obiektowego”
Informatyka i Ekonometria
II rok, stacjonarnie

Kraków, 21.01.2024.

1. Wstęp

W ramach projektu z przedmiotu "Programowanie obiektowe" miałyśmy za zadanie stworzyć aplikację do zarządzania sklepem z elektroniką. Celem projektu było zaprojektowanie i zaimplementowanie programu umożliwiającego użytkownikom zalogowanie się jako pracownik i zarządzanie produktami na stanie sklepu, przeglądanie produktów oraz katalogowaniem informacji o nich. Oprócz tego aplikacja umożliwia pracownikom dodawanie i usuwanie produktów z paragonu, a następnie dokonywanie zamówienia na potrzebne produkty.

2. Podział zadań

W trakcie realizacji projektu podzieliliśmy się zadaniami w następujący sposób:

1. Amelia Madej:
 - a. Zaimplementowała część klas i potrzebnych funkcjonalności .
 - b. Zastosowała interfejs IEquatable oraz własny interfejs IZapisywalnySklep.
 - c. Przeprowadziła testy jednostkowe dla wszystkich metod i konstruktorów.
 - d. Dokonała implementacji funkcji zapisu i odczytu danych z bazy danych SQL.
 - e. Stworzyła mechanizm pozwalający na tworzenie własnych wyjątków.
 - f. Zaimplementowała odczyt danych do/z pliku (serializacja XML)
2. Weronika Duda:
 - a. Zaimplementowała część klas i potrzebnych funkcjonalności .
 - b. Zaimplementowała funkcję korzystając z delegatów.
 - c. Stworzyła diagram UML dla klas występujących w systemie.
 - d. Zaimplementowała wykorzystanie interfejsów ICloneable i IComparable.
 - e. Wykonała sprawozdanie do projektu.
 - f. Stworzyła dokumentację techniczną.
3. Natalia Kargól:
 - a. Stworzyła rozbudowany interfejs graficzny (GUI), pozwalający na łatwą obsługę kluczowych elementów systemu.
 - b. Stworzyła kod obsługujący logikę interakcji z kontrolkami oraz logikę aplikacji (code-behind), wykorzystując bibliotekę klas „ElectronicStore”.
 - c. Dodała ruchomą animację w GUI.
 - d. Napisała instrukcję obsługi do projektu.

4. Opis klas w projekcie

a. Klasa abstrakcyjna Produkt

Klasa Produkt reprezentuje podstawową jednostkę dostępną w sklepie. Każdy produkt posiada unikalny identyfikator, producenta, model, cenę, opis oraz ilość dostępną na stanie sklepu.

b. Klasa Sklep

Klasa Sklep zarządza funkcjonalnością sklepu elektronicznego. Odpowiada za stan produktów, obsługę pracowników, realizację zakupów, a także zapisywanie i odczytywanie danych z pliku XML oraz bazy danych SQL.

c. Klasa Smartphone

Klasa Smartphone dziedziczy po klasie Produkt ma za zadanie reprezentować smartfony dostępne w sklepie. Dodatkowo, wprowadza specyficzne właściwości dotyczące parametrów technicznych smartfonów.

d. Klasa Laptop

Klasa Laptop również dziedziczy po klasie Produkt i reprezentuje laptopy, które posiadają unikalne właściwości, takie jak rozmiar ekranu, ilość pamięci RAM, pojemność dysku oraz system operacyjny (Windows, MacOS, Linux). Napisane metody „ToString()” oraz „WyswietlSzczegoly()” umożliwiają wyświetlanie informacji o laptopie.

e. Klasa Paragon

Klasa Paragon reprezentuje paragon zawierający listę produktów przeznaczonych do zakupu (zamówienia). Umożliwia ona dodawanie i usuwanie produktów z paragonu oraz wyświetlanie jego zawartość, a także sortowanie produktów według ceny oraz wyszukiwanie produktów wybranego producenta.

f. Klasa Osoba

Klasa Osoba stanowi abstrakcyjną bazę dla obiektów reprezentujących pracowników sklepu. Klasa ta zawiera podstawowe informacje o osobie, takie jak imię, nazwisko, miasto, data urodzenia, numer PESEL oraz płeć. Koncepcja zastosowania interfejsu IEquatable pozwala na efektywne porównywanie obiektów typu Osoba. Dodatkowo, klasa implementuje metodę „ToString()”, umożliwiającą wyświetlenie informacji o danej osobie.

g. Klasa Pracownik

Klasa Pracownik dziedziczy po klasie Osoba i dodatkowo zawiera informacje specyficzne dla pracowników sklepu, takie jak unikalny identyfikator, numer pracownika, login oraz hasło. Numer pracownika oraz login są generowane automatycznie przy tworzeniu nowego pracownika. Dodatkowo stworzona metoda „ToString()” umożliwia czytelne wyświetlanie informacji o pracowniku.

h. Interfejs IZapisywalnySklep

Interfejs IZapisywalnySklep definiuje jedną metodę: „ZapiszDoXml”, która umożliwia zapisanie obiektu implementującego interfejs do pliku XML.

i. Klasa SklepDbContext

Klasa SklepDbContext reprezentuje kontekst bazy danych dla aplikacji sklepu elektronicznego. Dziedziczy ona po klasie DbContext z Entity Framework. Klasa ta ma trzy właściwości reprezentujące tabele w bazie danych, czyli: Sklepy, Produkty oraz Pracownicy.

j. Klasa ZlyNumerPESELException

Klasa ZlyNumerPESELException (dziedzicząca ona po klasie bazowej Exception) ma za zadanie reprezentować wyjątek zgłaszany w przypadku wprowadzenia błędnego numeru PESEL.

k. Klasa AdminWindow

Klasa AdminWindow, to okno administratora (zalogowanego pracownika). Obsługa przycisków pozwala nawigować między różnymi funkcjonalnościami systemu np. zarządzaniem stanem magazynowym w sklepie, przyjmowaniem zamówień oraz uzyskiwaniem informacji o produktach. Przycisk „Logout” umożliwia wylogowanie się z konta administratora, przekierowując go do MainWindow.

l. Klasa MainWindow

Klasa MainWindow stanowi interfejs głównego okna logowania. W konstruktorze wczytuje dane sklepu z pliku XML, a następnie obsługuje logowanie użytkownika (jako administratora). Po wprowadzeniu nazwy użytkownika i hasła, program sprawdza, czy istnieje pracownik o podanych danych spośród pracowników sklepu. W przypadku poprawnego uwierzytelnienia, użytkownik zostaje przekierowywany do okna administratora (AdminWindow). Oprócz tego, interfejs pozwala za pomocą funkcji „pokaż hasło” wyświetlić wprowadzone hasło.

m. Klasa InventoryWindow

InventoryWindow to okno odpowiedzialne za zarządzanie stanem magazynowym w sklepie. Pozwala użytkownikowi na dodawanie nowych produktów, które są następnie zapisywane do pliku XML. Interfejs dynamicznie dostosowuje się do wybranej kategorii produktu (laptop czy smartphone), umożliwiając wprowadzenie specyficznych dla niej parametrów, takich jak rozmiar ekranu czy pojemność dysku, bądź aparat tylni/przedni.

n. Klasa TakeOrderWindow

TakeOrderWindow reprezentuje okno obsługujące proces składania zamówienia na potrzebne produkty. Umożliwia ono filtrowanie produktów według producenta i kategorii, a następnie dodawanie ich do paragonu. Interfejs pozwala na sortowanie pozycji w paragonie względem ceny za pomocą strzałek, a także usuwanie wybranych produktów. Po złożeniu zamówienia, stan magazynu jest aktualizowany. Okienko umożliwia również powrót do panelu administratora.

o. Klasa ProductInfoWindow

ProductInfoWindow to okno, które umożliwia sprawdzenie informacji o produktach dostępnych w sklepie. Po wyborze tego okienka użytkownik może (po wybraniu konkretnego produktu) zobaczyć zarówno podstawowe informacje o nim, jak i te szczegółowe. Dodana jest również opcja zobaczenia listy produktów dostępnych na stanie oraz (podobnie jak w innych oknach) kliknięcia strzałki oznaczającej powrót do panelu administratora.

p. ProduktsInStock

ProduktsInStock reprezentuje listę produktów dostępnych na stanie w sklepie. Informacje o dostępnych produktach pobierane są bezpośrednio z pliku XML, dzięki czemu użytkownik ma możliwość sprawdzenia aktualnej dostępności produktów w magazynie.

4. Posumowanie

Podczas realizacji poszczególnych zadań w projekcie zdobyliśmy cenne doświadczenie w programowaniu w języku C# oraz w pracy z bazami danych. W przyszłości istnieje wiele możliwości rozwoju projektu, takich jak dodanie funkcji dodawania pracowników, Implementacja interfejsu użytkownika dla zarządzania pracownikami, a także rozwinięcie funkcji bazodanowych. Jednak na ten moment uważamy, że nasz projekt spełnia założone cele i może być wykorzystany jako podstawa do rozbudowy i ulepszenia systemu zarządzania sklepem.