

Projekt PROI – sprawozdanie

1. Dane podstawowe

Autorzy:

Karol Kasperek

Szymon Dyszewski

Piotr Kosmala

Temat: Sklep z płytami winylowymi

2. Założenia

Realizowana przez nas symulacja ukazuje sposób działania sklepu z płytami winylowymi. W każdej jednostce czasu pojawiają się w nim klienci, którzy mają przy sobie listę zakupów (poszukiwane przez nich płyty, oraz potrzebna ilość). Kontaktują się oni ze sprzedawcami, po uprzednim odnalezieniu osoby odpowiedzialnej za sekcję sklepu zajmującą się gatunkiem muzycznym, do którego należy kupowana płyta. Gdy w magazynie jest wystarczający zapas, zamówienie klienta jest realizowane, zaś w przeciwnym wypadku sprzedawca informuje klienta o braku, a towar jest zamawiany.

3. Struktura programu

Kod podzielony jest na pięć klas oraz główny moduł testujący. Główny kontener danych stanowi klasa shop, która przechowuje listę wszystkich płyt winylowych, sprzedawców, sekcji oraz klientów. Klasa Section jest powiązana z klasą Vinyl, jako że przechowywane są w niej wszystkie winyle będące w danej sekcji. Obiekty klasy Seller są połączone z obiektami klasy Section odpowiadającymi sekcjom, za które są odpowiedzialni poszczególni sprzedawcy. Głównym celem klasy Client jest przeprowadzenie symulacji, zatem nie jest ściśle połączona z pozostałymi.

4. Testowanie programu

Przed wykonaniem kodu, użytkownik musi podać długość trwania symulacji oraz liczbę klientów, dla której zostanie przeprowadzona. Następnie rozpocznie zgodny z założeniami przebieg testu. Program wygeneruje losowo klientów, a następnie pokaże ich interakcje z personelem sklepu. Po dokonaniu zakupu zostanie obliczona i wyświetlona również kwota do zapłaty. Pętla odliczająca czas zatrzymuje się w każdej jednostce, czekając na pozwolenie użytkownika na kontynuację działania.

5. Biblioteka STL

W projekcie szeroko wykorzystywana jest struktura `vector` z biblioteki standardowej. Decyzja ta została podjęta przede wszystkim ze względu na łatwość dodawania kolejnych elementów na koniec kontenera, co jest bardzo częstą czynnością w naszym programie. Kolejną przydatną strukturą to `pair`, pozwalająca na przechowywanie dwóch zmiennych obok siebie. Nasz kod zawiera wektor par, dzięki któremu mogliśmy wygodnie działać na większej ilości zmiennych. W implementacji pojawiła się również struktura `map`.

6. Sytuacje wyjątkowe

Przebieg programu został zaprojektowany tak, aby wartości parametrów podawanych do metod były zawsze poprawne. Z tego powodu nasza uwaga skupiła się na weryfikacji wejścia użytkownika. Wykorzystany został do tego standardowy mechanizm obsługi wyjątków oparty o bloki `try/catch`, w których przechwytyjemy błąd `invalid_argument`, występujący w przypadku podania do terminala parametru niebędącego liczbą całkowitą.

7. Dalszy rozwój projektu

Stworzone funkcjonalności w pełni pokrywają wymagania projektu, jednak dodanie kilku z nich byłoby krytycznym warunkiem wykorzystania stworzonego programu. Jedną z nich jest „obsługiwanie” klientów, a więc po sprzedaniu płyt usunięcie jego zamówienia z wektora listy zakupów. Kolejną ważną funkcjonalnością byłoby zwalnianie statusu sprzedawców po pełnym zrealizowaniu listy zakupów klienta. Kolejne kroki powinny objąć rejestrację klientów oraz zarządzanie bazami danych – sprzedawcami i stanem magazynowym.