

## Dokumentacja projektu symulacji działania supermarketu.

### 1. Założenia projektowe

- 1.1. Supermarket powinien być klasą, która posiada kasy, produkty i pracowników. Klienci powinni mieć dostęp do produktów sklepu, kas oraz wchodzić w interakcję z pracownikami. Klasa kasa będzie zawierać w sobie potwierdzenie zapłaty (fakturę lub paragon), które będzie mogła przekazać klientowi przez pracownika.
- 1.2. Klient musi posiadać metodę, która pozwala na zadawanie pytań pracownikowi a także metodę sprawdzającą produkt.
- 1.3. Nazwy produktów będą zdefiniowane już w konstruktorze supermarketu.
- 1.4. Użytkownik będzie mógł zdefiniować liczbę iteracji symulacji, pracowników oraz kas poprzez wpisanie wartości w pliku, który program będzie odczytywał.
- 1.5. Przebieg symulacji zapisywany będzie do pliku oraz wypisywany na konsoli.

### 2. Hierarchia i relacje międzyklasowe

#### 2.1. Dziedziczenie:

- Klasa 'Product' jest klasą bazową dla klas: productVAT8 i productVAT23
- Klasa 'Payment' jest klasą bazową, po której dziedziczą Check i Invoice. Kolejno paragon jak i faktura są szczególnymi przypadkami potwierdzenia zapłaty.

#### 2.2. Zawieranie:

- Klasa 'Supermarket' zawiera w sobie kontenery (vectory) klasy cashRagister, Worker, productVAT8 i productVAT23.
- Klasa 'cashRagister' zawiera vector klasy Client.
- Klasa 'Client' zawiera vector klas productVAT8 i productVAT23.

#### 2.3. Inne:

- Klasa 'cashRegister' nie zawiera w sobie obiektu potwierdzenia płatności, natomiast posiada metodę umożliwiającą obliczanie jego elementów.

### 3. Opis symulacji

- 3.1. Inicjalizacja supermarketu poprzez wywołanie konstruktora z parametrami wczytanymi z pliku.
- 3.2. Jeśli sklep został otwarty, wchodzi do niego klient.
- 3.3. Wybór produktu przez klienta jest uzależniony od generatora liczb pseudolosowych i odbywa się dwa razy. W czasie wyboru może się zdarzyć, że klient sprawdzi cenę jakiegoś towaru.
- 3.4. Po wyborze produktów klient kieruje zapytanie do pracownika.
- 3.5. Pierwszy wolny pracownik otwiera pierwszą zamkniętą jeszcze kasę.
- 3.6. Klient wybiera kasę otwartą z najmniejszą kolejką.
- 3.7. Kasowanie klienta jest zaimplementowane razem z wyborem potwierdzenia zapłaty. Tylko w przypadku faktury pokaże się numer oraz VAT.
- 3.8. Po skasowaniu produktów klienta jest on usuwany z kolejki kasowej.

- 3.9. Przechodzimy do następnej iteracji.
- 3.10. Po zakończonych iteracjach, sklep jest zamykany.

#### 4. Wykorzystane biblioteki STL

- vector
- chrono
- iostream
- fstream
- random

#### 5. Obsługa wyjątków

- 5.1. Podczas kierowania pytań do pracowników przez klientów może się zdarzyć, że jest on zajęty (np. obsługa kasy). W funkcji pracownika, która zwraca jego odpowiedź, umieszczony został wyjątek, jeśli stan busy zwraca prawdę. Blok try i catch natomiast czeka na odpowiedź w funkcji pytającej klienta i powoduje to zwrócenie wiadomości „pracownik zajęty”.
- 5.2. Drugi przypadek zastosowania obsługi wyjątków umieszczony został przy wyborze rodzaju potwierdzenia płatności. Podczas gdy zostanie umieszczony inny rodzaj płatności, zostanie zgłoszony wyjątek o nieprawidłowości.