Dokumentacja

**Pakiet: Data**

**Klasa: Customer**

Opis: Reprezentuje klienta z danymi osobowymi, adresem, numerem telefonu, statusem stałego klienta i listą zamówień.

Atrybuty:

-name (String): Imię klienta.

-surname (String): Nazwisko klienta.

-sex (String): Płeć klienta.

-address (Address): Adres klienta.

-phoneNumber (String): Numer telefonu klienta.

-regularCustomer (String): Wskazuje, czy klient jest stałym klientem czy nie.

-orderList (Order): Lista zamówień klienta.

Konstruktory:

-Konstruktor bezargumentowy.

-Konstruktor parametrowy.

Metody:

-updatePhoneNumber(String newNumber): Aktualizuje numer telefonu klienta.

-updateCustomerAddress(Address newAddress): Aktualizuje adres klienta.

**Klasa: Invoice**

Opis: Reprezentuje fakturę z informacjami o kliencie, dacie faktury i liście produktów.

Atrybuty:

-customerName (String): Imię klienta.

-customerSurname (String): Nazwisko klienta.

-customerPhoneNumber (String): Numer telefonu klienta.

-invoiceDate (String): Data faktury.

-listOfProducts (List<Product>): Lista produktów na fakturze.

Konstruktory:

-Invoice(): Konstruktor domyślny.

-Invoice(Order order, Customer customer): Konstruktor parametrowy używający zamówienia i klienta do wypełnienia faktury.

Metody:

-sumInvoiceCost(): Oblicza całkowity koszt produktów na fakturze.

-generateInvoice(): Wyświetla szczegóły faktury.

**Klasa: Order**

Opis: Reprezentuje zamówienie z listą produktów, datą zamówienia i szczegółami płatności.

Atrybuty:

-listOfProducts (List<Product>): Lista produktów w zamówieniu.

-orderDate (String): Data zamówienia.

-paymentDetails (PaymentDetails): Szczegóły płatności za zamówienie.

Konstruktory:

-Order(List\<Product> listOfProducts): Konstruktor parametrowy z listą produktów.

Metody:

Brak dodatkowych metod.

**Klasa: PaymentDetails**

Opis: Reprezentuje szczegóły płatności, w tym metodę płatności i bonus.

Atrybuty:

-paymentMethod (String): Metoda płatności.

-bonus (int): Bonus związany z metodą płatności.

Konstruktory:

-Konstruktor bezargumentowy.

-Konstruktor parametrowy.

**Klasa: Product**

Opis: Reprezentuje produkt z nazwą, ceną i dostępną ilością.

Atrybuty:

-name (String): Nazwa produktu.

-price (int): Cena produktu.

-availableQuantity (int): Dostępna ilość produktu.

Konstruktory:

-Konstruktor bezargumentowy.

-Konstruktor parametrowy.

**Pakiet: Editors**

**Klasa: Constants**

Opis: Zawiera stałe wartości używane w aplikacji.

Atrybuty:

Różne tablice dla nazw ulic, miast, imion, nazwisk itp.

**Klasa: DataGenerator**

Opis: Generuje losowe dane, takie jak adresy, numery telefonów, klienci i zamówienia.

Metody:

Różne metody do generowania losowych danych, takich jak adresy, kody pocztowe, numery domów, typy klientów, metody płatności, ilości, ceny, daty zamówienia i produkty.

**Klasa: FileHelper**

Opis: Obsługuje operacje na plikach do przechowywania i odczytywania danych klientów.

Metody:

-saveCustomersToFile(List<Customer> customerList): Zapisuje dane klientów do pliku.

-readFromFile(): Odczytuje dane z pliku i zwraca listę ciągów znaków.

**Klasa: StatisticsGenerator**

Opis: Generuje statystyki na podstawie danych klientów.

Metody:

-calculatePurchaseCostForCustomer(List<Customer> customerList): Oblicza i wyświetla łączny koszt zakupów dla każdego klienta.

-calculateAverageCostsBySex(List<Customer> customerList): Oblicza i wyświetla średni koszt zakupów w zależności od płci klienta.

**Pakiet: Tests**

**Klasa: TestGenerator**

Opis: Klasa zawierająca testy jednostkowe dla metod klasy DataGenerator.

Metody:

-testGenerateQuantity(): Testuje, czy metoda generateQuantity() klasy DataGenerator zwraca liczbę nieujemną.

-testGeneratePhoneNumber(): Testuje, czy metoda generatePhoneNumber() klasy DataGenerator zwraca poprawny format numeru telefonu (XXX-XXX-XXX).

-testGeneratePostalCode(): Testuje, czy metoda generatePostalCode() klasy DataGenerator zwraca poprawny format kodu pocztowego (XX-XXX).

-testGeneratePrice(): Testuje, czy metoda generatePrice() klasy DataGenerator zwraca cenę nieujemną.

-testGenerateAddress(): Testuje, czy metoda generateAddress() klasy DataGenerator zwraca niepusty adres.