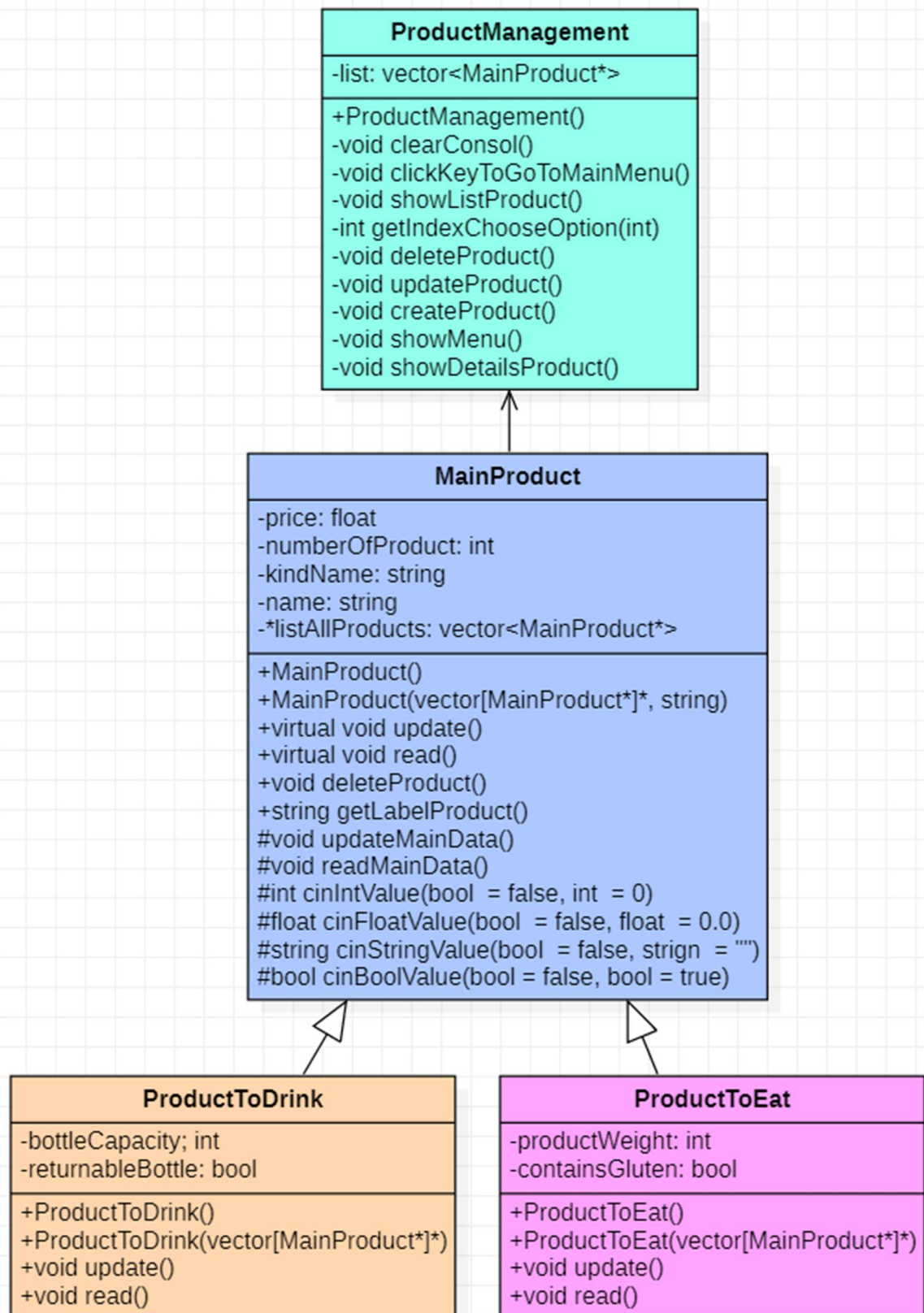


# Projekt bazy danych sklepu spożywczego w C++



## 1. Klasa **ProductManagement**:

- a. Zastosowanie
  - i. Służy do zarządzania bazą danych. Ta klasa jest odpowiedzialna za wyświetlanie menu programu, pokazywanie listy dostępnych w bazie produktów, wybieranie który produkt usunąć, jaki produkt dodać oraz jaki produkt zaktualizować.
- b. Zmienne klasy
  - i. List – przechowuje tablicę wektorów z dostępnymi w bazie produktami
- c. Metody Publiczne
  - i. ProductManagement() – tworzy obiekt i wyświetla menu
- d. Metody Prywatne
  - i. void clearConsole() – czyści okienko konsoli
  - ii. void clickKeyToGoToMainMenu() – po wywołaniu funkcji program czeka na wciśnięcie dowolnego programu, aby przejść do głównego menu
  - iii. void showListProduct() – funkcja wypisuje wszystkie dostępne produkty
  - iv. int getIndexChooseOption(int max) – funkcja zwraca wpisaną liczbę, całkowitą, ale czeka aż użytkownik wpisze nieujemną i mniejszą od max
  - v. void deleteProduct() – funkcja wyświetla listę produktów i czeka aż użytkownik wybierze który produkt ma zostać usunięty i uruchamia metodę odpowiadającą za usunięcie w klasie produktu
  - vi. void updateProduct() – funkcja wyświetla listę produktów i czeka, aż użytkownik wybierze produkt do zaktualizowania i uruchamia metodę odpowiadającą za update w klasie produktu
  - vii. void createProduct() - funkcja wyświetla listę kategorii produktów i czeka, aż użytkownik wybierze kategorię produkt, następnie uruchamia metodę odpowiadającą za tworzenie w klasie
  - viii. void showMenu() – pokazuje główne menu programu
  - ix. void showDetailsProduct() - funkcja wyświetla listę produktów i czeka, aż użytkownik wybierze, który produkt ma zostać szczegółowo pokazany i uruchamia metodę odpowiadającą za pokazanie szczegółów w klasie produktu

## 2. Klasa **MainProduct**

- a. Zastosowanie
  - i. Klasa jest główną częścią każdego rodzaju produktu w sklepie (każda klasa produktu dziedziczy z tej klasy). Ta klasa przechowuje informacje, które są stałe dla wszystkich produktów
- b. Zmienne klasy
  - i. price – cena podawana jako liczba zmiennoprzecinkowa
  - ii. numberOfProduct – liczba produktów w magazynie
  - iii. kindName – nazwa kategorii do której należy produkt
  - iv. name – nazwa produktu
  - v. \*listAllProducts – wskaźnik do listy wszystkich produktów
- c. Metody Publiczne
  - i. MainProduct()
  - ii. MainProduct(vector<MainProduct\*>\*, string) – konstruktor klasy, który pobiera od użytkownika dane dotyczące głównych informacji o produkcie
  - iii. virtual void update() – wirtualna metoda aktualizowania, jest przesłaniana w klasach pochodnych
  - iv. virtual void read() – wirtualna metoda wypisywania szczegółów produktu, jest przesłaniana w klasach pochodnych
  - v. void deleteProduct() – funkcja usuwa produkt z listy dostępnych produktów
  - vi. string getLabelProduct() – funkcja zwraca napis z nazwą produktu i jego kategorią
- d. Metody Chronione

- i. void updateMainData() – funkcja wywoływana przez klasy pochodne podczas aktualizowania danych produktu, aby zaktualizować główne dane produktu
- ii. void readMainData() - funkcja wywoływana przez klasy pochodne podczas aktualizowania danych produktu, aby wypisać główne dane produktu
- iii. int cinIntValue(bool = false, int = 0) – funkcja używana w celu wymuszenia od użytkownika żeby wpisał liczbę całkowitą dodatnią i funkcja ją zwraca, jeżeli bool=true to może użyć samego entera i zostanie zwrócona wartość int
- iv. float cinFloatValue(bool = false, float = 0.0) – funkcja używana w celu wymuszenia od użytkownika żeby wpisał liczbę zmiennoprzecinkową dodatnią i funkcja ją zwraca, jeżeli bool=true to może użyć samego entera i zostanie zwrócona wartość float
- v. string cinStringValue(bool = false, strign = "") - funkcja używana w celu wymuszenia od użytkownika żeby wpisał napis i funkcja ją zwraca, jeżeli bool=true to może użyć samego entera i zostanie zwrócona wartość string
- vi. bool cinBoolValue(bool = false, bool = true) - funkcja używana w celu wymuszenia od użytkownika żeby wpisał napis „0” lub „1” i funkcja zwraca odpowiednio false lub true, jeżeli bool=true to może użyć samego entera i zostanie zwrócona wartość bool

### 3. Klasa **ProductToDrink**

- a. Zastosowanie
  - i. Klasa służy do tworzenia i zarządzania produktem z kategorii napój, dziedziczy z klasy MainProduct
- b. Zmienne klasy
  - i. bottleCapacity – wartość całkowita oznaczająca pojemność butelki w mililitrach
  - ii. returnableBottle – wartość boolowska określająca czy butelka jest zwrotna
- c. Metody Publiczne
  - i. ProductToDrink()
  - ii. ProductToDrink(vector[MainProduct]\*) – wywołuje konstruktor MainProduct(vector[MainProduct]\*, string) oraz dodatkowa pobiera od użytkownika informacje specyficzne dla kategorii napój
  - iii. void update() – funkcja wywołuje updateMainData() oraz aktualizuje dane specyficzne dla kategorii napój
  - iv. void read() – funkcja wywołuje readMainData() oraz wypisuje dane specyficzne dla kategorii napój

### 4. Klasa **ProductToEat**

- a. Zastosowanie
  - i. Klasa służy do tworzenia i zarządzania produktem z kategorii jedzenie, dziedziczy z klasy MainProduct
- b. Zmienne klasy
  - i. productWeight – wartość całkowita oznaczająca wagę produktu w gramach
  - ii. containsGluten – wartość boolowska określająca czy produkt zawiera gluten
- c. Metody Publiczne
  - i. ProductToEat ()
  - ii. ProductToEat (vector[MainProduct]\*) – wywołuje konstruktor MainProduct(vector[MainProduct]\*, string) oraz dodatkowa pobiera od użytkownika informacje specyficzne dla kategorii jedzenie
  - iii. void update() – funkcja wywołuje updateMainData() oraz aktualizuje dane specyficzne dla kategorii jedzenie
  - iv. void read() – funkcja wywołuje readMainData() oraz wypisuje dane specyficzne dla kategorii jedzenie