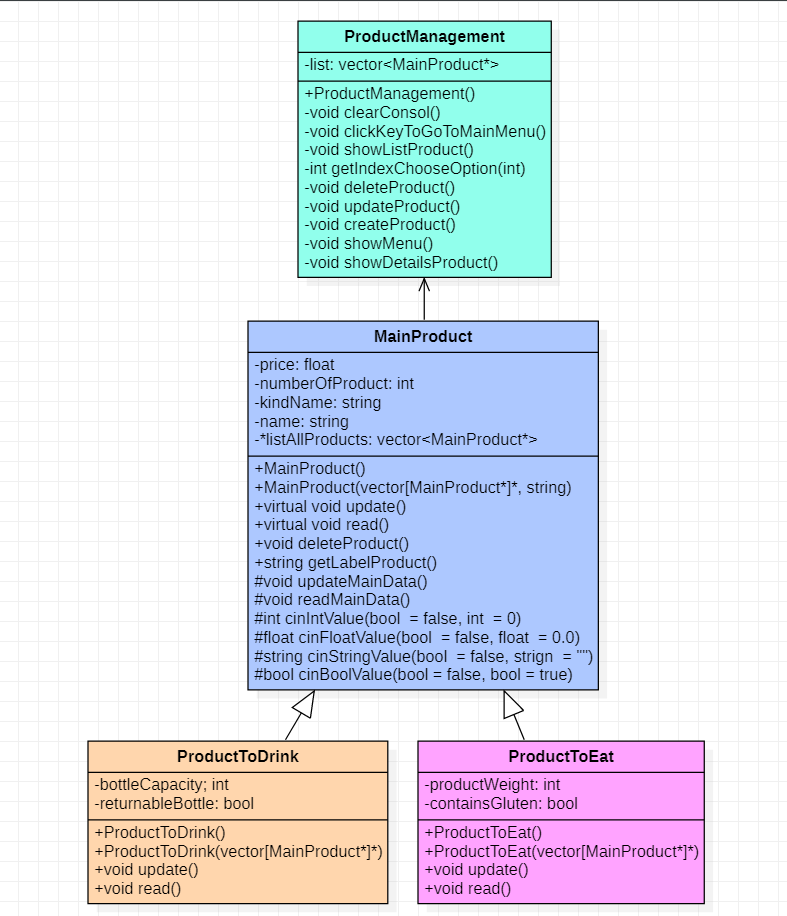
# Patryk Lukaszewski 148058 I6.1

**Projekt bazy danych sklepu spożywczego w C++**

1. Klasa **ProductManagement**:
   1. Zastosowanie
      1. 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.
   2. Zmienne klasy
      1. List – przechowuje tablicę wektorów z dostępnymi w bazie produktami
   3. Metody Publiczne
      1. ProductManagement() – tworzy obiekt i wyświetla menu
   4. Metody Prywatne
      1. void clearConsol() – czyście okienko konsoli
      2. void clickKeyToGoToMainMenu() – po wywołaniu funkcji program czeka na wciśnięcie dowolnego programu, aby przejść do głównego menu
      3. void showListProduct() – funkcja wypisuje wszystkie dostępne produkty
      4. int getIndexChooseOption(int max) – funkcja zwraca wpisaną liczbę, całkowitą, ale czeka aż użytkownik wpisze nieujemną i mniejszą od max
      5. 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
      6. void updateProduct() – funkcja wyświetla listę produktów i czeka, aż użytkownik wybierze produkt do zaktualizowania i uruchamia metodę odpowiadającą za update w klasie produktu
      7. void createProduct() - funkcja wyświetla listę kategorii produktów i czeka, aż użytkownik wybierze kategorię produkt, następnie uruchamia metodę odpowiadającą za tworzenie w klasie
      8. void showMenu() – pokazuje główne menu programu
      9. 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**
   1. Zastosowanie
      1. 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
   2. Zmienne klasy
      1. price – cena podawana jako liczba zmiennoprzecinkowa
      2. numberOfProduct – liczba produktów w magazynie
      3. kindName – nazwa kategorii do której należy produkt
      4. name – nazwa produktu
      5. \*listAllProducts – wskaźnik do listy wszystkich produktów
   3. Metody Publiczne
      1. MainProduct()
      2. MainProduct(vector<MainProduct\*>\*, string) – konstruktor klasy, który pobiera od użytkownika dane dotyczące głównych informacji o produkcie
      3. virtual void update() – wirtualna metoda updatowania, jest przesłaniana w klasach pochodnych
      4. virtual void read() – wirtualna metoda wypisywania szczegółów produktu, jest przesłaniana w klasach pochodnych
      5. void deleteProduct() – funkcja usuwa produkt z listy dostępnych produktów
      6. string getLabelProduct() – funkcja zwraca napis z nazwą produktu i jego kategorią
   4. Metody Chronione
      1. void updateMainData() – funkcja wywoływana przez klasy pochodne podczas aktualizowania danych produktu, aby zaktualizować główne dane produktu
      2. void readMainData() - funkcja wywoływana przez klasy pochodne podczas aktualizowania danych produktu, aby wypisać główne dane produktu
      3. 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
      4. 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
      5. 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
      6. 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**
   1. Zastosowanie
      1. Klasa służy do tworzenia i zarządzania produktem z kategorii napój, dziedziczy z klasy MainProduct
   2. Zmienne klasy
      1. bottleCapacity – wartość całkowita oznaczająca pojemność butelki w mililitrach
      2. returnableBottle – wartość boolowska określająca czy butelka jest zwrotna
   3. Metody Publiczne
      1. ProductToDrink()
      2. ProductToDrink(vector[MainProduct\*]\*) – wywołuje konstruktor MainProduct(vector[MainProduct\*]\*, string) oraz dodatkowa pobiera od użytkownika informacje specyficzne dla kategorii napój
      3. void update() – funkcja wywołuje updateMainData() oraz aktualizuje dane specyficzne dla kategorii napój
      4. void read() – funkcja wywołuje readMainData() oraz wypisuje dane specyficzne dla kategorii napój
4. Klasa **ProductToEat**
   1. Zastosowanie
      1. Klasa służy do tworzenia i zarządzania produktem z kategorii jedzenie, dziedziczy z klasy MainProduct
   2. Zmienne klasy
      1. productWeight– wartość całkowita oznaczająca wagę produktu w gramach
      2. containsGluten – wartość boolowska określająca czy produkt zawiera gluten
   3. Metody Publiczne
      1. ProductToEat ()
      2. ProductToEat (vector[MainProduct\*]\*) – wywołuje konstruktor MainProduct(vector[MainProduct\*]\*, string) oraz dodatkowa pobiera od użytkownika informacje specyficzne dla kategorii jedzenie
      3. void update() – funkcja wywołuje updateMainData() oraz aktualizuje dane specyficzne dla kategorii jedzenie
      4. void read() – funkcja wywołuje readMainData() oraz wypisuje dane specyficzne dla kategorii jedzenie