**Zadanie 1)**

1. Stwórz klasę **Blad**, a w niej konstruktor bezargumentowy oraz metodę *Write()* wypisującą na ekran nazwę klasy.
2. Stwórz klasę **DzieleniePrzezZero**, pochodną klasy **Blad**, a w niej konstruktor bezargumentowy oraz metodę *Write()* wywołującą metodę *Write()* z klasy bazowej, a także wypisującą na ekran nazwę klasy.
3. Stwórz klasę **BladObslugiPliku**, pochodną klasy **Blad**, a w niej konstruktor bezargumentowy oraz metodę *Write()* wypisującą na ekran nazwę klasy.

**Zadanie 2)**

Otwórz plik wejściowy z liczbami zmiennoprzecinkowymi; wykorzystując zagnieżdżenie wyrzuć oraz złap wyjątek typu **BladObslugiPliku**, na obydwu poziomach zagnieżdżenia wypisz odpowiedni komunikat, informujący o tym poziomie, a także wywołaj metodę *Write()* dla złapanego obiektu.

**Zadanie 3)**

Stwórz listę jednokierunkową liczb zmiennoprzecinkowych typu **Element** i dodaj do niej liczby z

pliku wejściowego. Jeżeli nie uda się zaalokować pamięci, złap wyjątek używając wyrażenia

*bad\_alloc.*

**Zadanie 4)**

Napisz funkcję *Podziel*(**Element** \**el*, **double** *n*) zmieniającą wartości liczb w stworzonej liście, dzieląc każdy element przez liczbę *n* podaną przez użytkownika. Jeżeli liczba ta będzie równa 0, rzuć wyjątek typu **DzieleniePrzezZero** oraz złap go, używając dwa razy funkcji *catch* – w odniesieniu do klasy **DzieleniePrzezZero** oraz do klasy **Blad**. Pamiętaj o odpowiedniej kolejności wywołania tych funkcji. Poinformuj o typie rzucanego wyjątku, używając metody *Write().*