Zad. 7

Zadanie polega na napisaniu 2 klas, które będą reprezentowały planety i inne ciała niebieskie. W tym celu proszę wykorzystać mechanizm dziedziczenia.

Pierwsza klasa to *CialoNiebieskie*, która ma następujące pola *protected* lub *private*:
- float x – pozycja x w galaktyce
- float y – pozycja y w galaktyce
- float m – masa
- string nazwa
- float Vx – prędkość w kierunku x
- float Vy – prędkość w kierunku x

CialoNiebieskie posiada również konstruktor domyślny i konstruktor główny. W konstruktorze głównym pola są inicjalizowane za pomocą listy inicjalizacyjnej.

Dla klasy *CialoNiebieskie* należy zaimplementować:
- publiczną metodę *void Ruch(float dt)*, która wykonywałaby symulację ruchu planety/gwiazdy w galaktyce. Ta metoda oblicza nową pozycję ciała niebieskiego na podstawie aktualnej pozycji, prędkości i kroku czasowego *dt*. Na potrzeby tego zadania robimy naiwne założenie, że dane ciało porusza się ruchem jednostajnym.
- mechanizm wyświetlania informacji o obiekcie: albo metodę info(), wyświetlającą wszystkie informacje, albo przeciążony operator „<<”.

Klasa *Planeta* dziedziczy publicznie po klasie *CialoNiebieskie* i ma dodatkowo dwa pola prywatne:
- *int nSatelitów* – ilość satelitów krążących wokół planety
- *CialoNiebieskie* lista* - tablica dynamiczna, w której przechowywane są informacje o satelitach krążących wokół planet. Domyślny rozmiar tablicy to 100.

Dla klasy *Planeta* proszę napisać konstruktor domyślny oraz destruktor. Proszę również zaimplementować:
- metodę *int DodajSatelite*(*CialoNiebieskie& ksiezyc*), która dodaje *księżyc* do listy posiadanych przez planetę satelitów, wpisując go na pierwszej wolnej pozycji w tablicy *lista*, uaktualnia wartość *nSatelitów*, a w końcu zwraca aktualną wartości *nSatelitów*.
- metodę *void Ruch(float dt)*, która oblicza nową pozycję planety oraz wszystkich jej satelitów.
- mechanizm wyświetlania informacji o obiekcie: albo metodę info(), wyświetlającą wszystkie informacje, albo przeciążony operator „<<”.
- przeciążony operator przypisania „=”

Do klas można dodawać dowolne pola i metody, wedle potrzeb i przyjętej koncepcji.

W programie głównym należy:
1. Stworzyć tablicę *CialoNiebieskie galaktyka[5];*
2. Wpisać do tablicy *galaktyka* obiekty reprezentujące: *CialoNiebieskie* Słońce, *CialoNiebieskie* Merkury, *CialoNiebieskie* Wenus, *Planeta* Ziemia, *Planeta* Mars. Wartości położeń i prędkości tych ciał są dowolne.
3. Stworzyć tablicę *CialoNiebieskie* galaktykaWsk[5];*
4. Do tablicy *galaktykaWsk* wpisać te same obiekty jak w *galaktyka*, korzystając z konstruktorów kopiujących.
5. Wyświetlić informacje o obiektach w tablicy *galaktyka*;
6. Wyświetlić informacje o obiektach w tablicy *galaktykaWsk*;
7. Wyświetlić informację o Marsie.
8. Zwolnić pamięć.

Punktacja:

- implantacja klas – 2.5 pkt
- wykorzystanie mechanizmów dziedziczenia – 1.5 pkt
- implementacja programu głównego – 1 pkt
- wykorzystanie klasy vector w funkcji *main* lub klasie *Planeta* – 1pkt

Program powinien zostać napisany z podziałem na 3 pliki: 2 pliki dla klasy (.h i .cpp) oraz oddzielny plik dla funkcji *main*.