

Zadanie 1 (na 3.0). Stwórz klasy:

- Obiekt materialny, posiadający pozycję (wektor) i prędkość (wektor),
- Ławka, dziedzicząca po obiekcie materialnym, posiadająca kolor (*enum*) oraz poziom wygody (*enum*),
- Taczka, dziedzicząca po obiekcie materialnym, posiadająca nośność (liczba rzeczywista).

Klasa obiektu materialnego ma dodatkowo posiadać:

- funkcję do pobrania pozycji,
- funkcję do pobrania prędkości,
- funkcję do zmiany pozycji,
- funkcję do zmiany prędkości,
- funkcję wirtualną wypisującą opis obiektu (np. „to jest obiekt materialny”),
- konstruktor zerujący pozycję i prędkość.

Klasa ławki ma dodatkowo posiadać:

- funkcję do pobrania koloru,
- funkcję do pobrania wygody,
- konstruktor ustawiający pozycję, kolor i wygodę na wartości przesłane do niego,
- nadpisaną funkcję wypisującą opis obiektu (np. „to jest ławka w kolorze X o wygodzie Y”).

Klasa taczki ma dodatkowo posiadać:

- funkcję do pobierania nośności,
- konstruktor ustawiający nośność na wartość przesłaną do niego,
- nadpisaną funkcję wypisującą opis obiektu (np. „to jest taczka o nośności X”).

Stwórz dwa wskaźniki na obiekt materialny, do jednego przypisz obiekt ławki, do drugiego taczki. Zaprezentuj działanie funkcji wypisującej opis.

Zadanie 2 (na 4.0). Po niezliczonych godzinach rozmyślania nad bluźnierczą, nieeuklidesową geometrią, Zły Markiz Cul de Sac wpadł na genialny pomysł – jego maszyna dostanie potężne działa strzelające trwałymi leptonami o spinie $\frac{1}{2}$ i ładunku równym $-e$. Stwórz klasy:

- Obiekt strzelający, posiadający jedną funkcję czysto wirtualną „wystrel” (tzw. interfejs),
- Działo, dziedziczące po obiekcie materialnym i obiekcie strzelającym, posiadające moc (liczba rzeczywista), nadpisaną funkcję wypisującą opis obiektu oraz konstruktor ustawiający moc na wartość przesłaną do niego,
- Działo strzelające seriami, implementujące funkcję „wystrel”,
- Działo strzelające pojedynczymi pociskami, implementujące funkcję „wystrel”.

Dwie ostatnie klasy działają mają nadpisać funkcję wypisującą opis obiektu, oraz wywoływać wersję z klasy nadrzędnej.

Strzelanie może polegać na wypisaniu „Bzzzt!” na ekran odpowiednią liczbę razy. Zaprezentuj działanie obu klas.

Zadanie 3 (na 5.0). Finalnie, nadszedł czas na złożenie części w idealną broń zagłady:

Taczko-Ławkę

Stwórz klasę Taczko-Ławki, dziedziczącą po klasie ławki i klasie taczki, posiadającą:

- dwa (wskaźniki na) działa leptonowe,
- funkcję wystrzeliwującą z obu dział,
- konstruktor wywołujący wszystkie konstruktory klas nadrzędnych, przyjmujący jako argumenty dwa wskaźniki na działa leptonowe,
- destruktor zwalniający pamięć,
- nadpisaną funkcję wypisującą opis obiektu (np. „To jest Ławko-Tacz-- Taczko-Ławka!”).

Stwórz obiekt Taczko-Ławki z jednym działem leptonowym strzelającym pojedynczo i jednym strzelającym seriami, przestaw pozycję i prędkość Taczko-Ławki, wystrzel, a na koniec wypisz jej pozycję. Stwórz wskaźnik na obiekt materialny, przypisz do niego obiekt Taczko-Ławki, a następnie go usuń (*delete*).

Zwróć uwagę, że Taczko-Ławka dwukrotnie dziedziczy po obiekcie materialnym (przez ławkę i przez taczkę). Co zostanie wypisane w poniższym przypadku?

```
TaczkoLawka *tl=new TaczkoLawka(new DzialoSeria(5), new DzialoJeden(2));
printf("Taczko-Lawka: %f %f %f\n",
      tl->getPosition().x, tl->getPosition().y, tl->getPosition().z);
Taczka *t=(Taczka*)tl;
printf("Taczka: %f %f %f\n",
      t->getPosition().x, t->getPosition().y, t->getPosition().z);
Lawka *l=(Lawka*)tl;
printf("Lawka: %f %f %f\n",
      l->getPosition().x, l->getPosition().y, l->getPosition().z);
```

Epilog (bez oceny). Zadowolony z efektywności prototypu, Zły Markiz Cul de Sac postanowił uruchomić produkcję masową. Rozgrzał linię produkcyjną i stworzył sto idealnych kopii prototypu Taczko-Ławki. Nie wiedział jednak, że czeka go niemiła niespodzianka.

```
TaczkoLawka tl(new DzialoSeria(5), new DzialoJeden(2));
TaczkoLawka armia[100];
for(int i=0; i<100; ++i)
    armia[i]=tl;
```

Co się stanie po zakończeniu programu?

Ad. 1

https://pl.wikipedia.org/wiki/Wyliczeniowy_typ_danych

Ad. 2

https://en.wikibooks.org/wiki/More_C%2B%2B_Idioms/Interface_Class

Ad. 3

https://en.wikipedia.org/wiki/Virtual_inheritance