

ELEMENTY JĘZYKA C++: dziedziczenie, klasy abstrakcyjne, funkcje wirtualne.

1. Macierzowa optyka geometryczna

W optyce macierzowej każdy układ elementów optycznych ustawionych wzdłuż jednej osi opisany jest pewną macierzą 2×2 . W szczególności pusty odcinek o długości d oraz cienka soczewka o ogniskowej f reprezentowane są odpowiednio macierzami:

$$S = \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix} \quad L = \begin{bmatrix} 1 & 0 \\ -1/f & 1 \end{bmatrix}$$

Jeżeli na osi optycznej bezpośrednio za elementem opisanym macierzą M_1 stoi element opisany macierzą M_2 , to układowi temu odpowiada macierz $M = M_2 M_1$. Promień światła opisany jest dwuwymiarowym wektorem

$$u = \begin{bmatrix} y \\ \alpha \end{bmatrix}$$

gdzie y oraz α są odpowiednio odległością promienia od osi optycznej oraz kątem między promieniem a osią. Promień opisany wektorem u_0 po przejściu przez układ opisany macierzą M przechodzi w promień opisany wektorem $u_1 = M u_0$.

Napisz klasę **Ray** reprezentującą promień światła. Zaimplementuj w niej: 1) bezargumentowy konstruktor zerujący y oraz α ; 2) dwuargumentowy konstruktor inicjalizujący y oraz α swoimi argumentami; 3) funkcje składowe `getDistance` oraz `getAngle` zwracające odpowiednio y i α .

Napisz klasę **System** opisującą układ elementów optycznych. Zaimplementuj w niej bezargumentowy konstruktor inicjalizujący macierz układu macierzą jednostkową. Jako funkcję zaprzyjaźnioną z klasą **System** zaimplementuj operator `*` odpowiadający składaniu dwóch układów ustawionych bezpośrednio jeden za drugim. Jako funkcję zaprzyjaźnioną z obiema klasami zaimplementuj operator `*` odpowiadający działaniu układu na promień.

Z klasy **System** wywiedź klasę **Space** reprezentującą pusty odcinek osi optycznej. Zaimplementuj w niej jednoargumentowy konstruktor inicjalizujący długość odcinka swoim argumentem. Z klasy **System** wywiedź też klasę **Lens** reprezentującą soczewkę. Zaimplementuj w niej jednoargumentowy konstruktor inicjalizujący ogniskową swoim argumentem.

Wszystkie klasy i funkcje powinny być napisane tak, aby działał następujący program:

```
int main(){
    Ray ray = Lens(50) * Space(150) * Lens(100) * Ray(0.1, 0);
    cout << ray.getAngle() << " " << ray.getDistance() << endl;
    return 0;}
```

2. Definicja dowolnego ciągu

Zadeklaruj abstrakcyjną klasę `Sequence` reprezentującą ciąg matematyczny. Zadeklaruj w niej publiczny operator indeksowania zwracający żądany wyraz ciągu. Z klasy tej wyprowadź klasę `Arithmetic` reprezentującą ciąg arytmetyczny: $a_n = a_0 + n\Delta$. Zdefiniuj dla niej konstruktor umożliwiający nadanie wartości parametrom a_0 oraz Δ . Z klasy `Sequence` wyprowadź również klasę `Fibonacci` reprezentującą ciąg Fibonacciego. Klasy te powinny być tak napisane, aby można je było wykorzystać w programie:

```
int main(){
    Sequence arithmetic( 2., 3. );
    Sequence fibonacci;
    for ( int n = 0; n < 10; ++n ){
        cout << arithmetic[ n ] << " ";
    }
    cout << endl;
    for ( int n = 0; n < 10 ; ++n ){
        cout << fibonacci[ n ] << " ";
    }
    cout << endl;}
```

3. Pochodna dowolnej funkcji

Zadeklaruj abstrakcyjną klasę `Function` reprezentującą funkcję rzeczywistą zmiennej rzeczywistej. Zadeklaruj w niej wirtualny operator wywołania funkcji, który umożliwia obliczenie wartości dowolnej pochodnej funkcji w dowolnym punkcie. Jeżeli rząd pochodnej nie jest podany, to operator powinien zwracać wartość samej funkcji. Z klasy `Function` wyprowadź klasy `Exponential` oraz `Sine` reprezentujące odpowiednio funkcje $\exp(\alpha x)$ oraz $\sin(\omega x)$. Zaimplementuj w nich konstruktory inicjalizujące wartości stałych α i ω oraz konkretyzacje operatora wywołania funkcji odziedziczony z klasy `Function`. Wymienione klasy skonstruuj w taki sposób, aby można je było wykorzystać w programie:

```
int main(){
    Exponential exponential( 2.);
    Sine sine( 3.);
    Function& expo = exponential;
    Function& sinus = sine;
    cout << expo( 1. ) << endl;
    cout << sinus( 2. , 1 ) << endl;}
```

Pytania, a także rozwiązania zadań, można wysyłać na adres: MDABROWSKI@FUW.EDU.PL.