Programowanie II R

Zadania – seria 2.

Funkcje

Zadanie 1: Podstawy funkcji i przekazywanie argumentów przez wartość

Napisz program obliczający energię kinetyczną obiektu w mechanice klasycznej oraz relatywistycznej.

1. Zaimplementuj funkcję `double energiaKlasyczna(double masa, double predkosc)`, zwracającą: $E_k = \frac{1}{2}mv^2$. Przyjmij, że masa podana jest w kg, a prędkość w m/s.
2. Zaimplementuj funkcję `double energiaRelatywistyczna(double masa, double predkosc)`, korzystając ze wzoru:
$$E_k = mc^2 \left(\frac{1}{\sqrt{1 - \frac{v^2}{c^2}}} - 1 \right)$$
Przyjmij prędkość światła $c = 299\,792\,458$ m/s.
3. W funkcji `main` pobierz masę od użytkownika i wyświetl różnicę wyników dla $v = 0.1c$ oraz $v = 0.9c$.

Wskazówki: Zdefiniuj prędkość światła jako zmienną wewnątrz funkcji, której wartość znana jest w momencie kompilacji: `constexpr double c = ...;`.

Zadanie 2: Organizacja projektu (Nagłówki i pliki źródłowe)

Rozbij projekt z Zadania 1 na trzy osobne pliki:

- `fizyka.hpp`: Zawiera deklaracje (prototypy) funkcji.
- `fizyka.cpp`: Zawiera pełne definicje funkcji.
- `main.cpp`: Zawiera funkcję `main`.

Wskazówki: Zabezpiecz plik nagłówkowy przed wielokrotnym dołączeniem za pomocą dyrektywy preprocesora `#pragma once`.

Zadanie 3: Przekazywanie argumentów przez referencje

Napisz funkcję `aktualizujStan`, która modeluje ruch jednostajny metodą Eulera:

$$x^{t+1} = x^t + v^t \Delta t$$

1. Funkcja powinna przyjmować przez **referencję** zmienną `x` (położenie), a przez wartość: `v` (prędkość) oraz `dt` (krok czasowy).
2. W `main` przyjmij od użytkownika: położenie początkowe, prędkość obiektu i długość symulacji. Symuluj ruch przez zadany czas poprzez wielokrotne wywołanie funkcji `aktualizujStan`. Przyjmij krok czasowy $\Delta t = 0.01$. Wypisz końcowe położenie i prędkość.

Wskazówki: Przykładowa deklaracja funkcji przyjmującej jeden argument przez referencję: `void fun(double &a)`.

Zadanie 4: Przeciążanie funkcji i domyślne wartości argumentów

Do programu z zadania 3. dodaj funkcję `aktualizujStan`, która modeluje ruch przyspieszony metodą Eulera:

$$x^{t+1} = x^t + v^t \Delta t$$
$$v^{t+1} = v^t + a^t \Delta t$$

1. Funkcja powinna przyjmować przez **referencję** zmienne `x` (położenie) oraz `v` (prędkość), a przez wartość: `a` (przyspieszenie) oraz `dt` (krok czasowy).
2. W `main` przyjmij od użytkownika: położenie początkowe, prędkość obiektu, przyspieszenie i długość symulacji. Symuluj ruch przez zadany czas poprzez wielokrotne wywołanie funkcji `aktualizujStan`. Przyjmij krok czasowy $\Delta t = 0.01$. Wypisz końcowe położenie i prędkość.
3. W obydwu funkcjach `aktualizujStan` ustaw domyślną wartość argumentu dt=0.01. Dlaczego program się nie kompiluje?

Wskazówki: Przykładowa deklaracja funkcji przyjmującej jeden argument z wartością domyślną: `void fun(double a = 10)`.

Zadanie 5: `std::function`

Często przydatne jest przekazywanie funkcji jako argumentu innej funkcji. Można to zrobić używając obiektu `std::function`. Przykładowo, `std::function<double(double)>` to typ, który reprezentuje funkcję przyjmującą jeden argument typu `double` i zwracającej również `double`.

Celem zadania jest implementacja całkowania numerycznego metodą prostokątów dla funkcji jednej zmiennej.

Opis metody

Dzielimy przedział $[a, b]$ na n równych podprzedziałów o szerokości $\Delta x = \frac{b-a}{n}$. W każdym podprzedziale wybieramy punkt środkowy x_i^* i przyjmujemy, że wartość funkcji w tym punkcie jest wysokością prostokąta. Pole całki to suma pól wszystkich n prostokątów:

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n-1} f(x_i^*)\Delta x \tag{1}$$

Zadanie

1. Zdefiniuj funkcję `calkuj(double a, double b, int n, std::function<double(double)> f)`, która całkuje funkcję f na przedziale [a,b] używając n podprzedziałów.
2. Użyj jej do scałkowania funkcji `sin` na przedziale $[0, \pi]$, używając 10, 100 i 1000 podprzedziałów.

Wskazówki: `std::function` i `std::sin` zdefiniowane są w plikach nagłówkowych `functional` i `cmath`.

Zadanie 6: Wyrażenia lambda (funkcje anonimowe)

Napisz program, który używa funkcji `całkuj` z Zadania 5 do scałkowania funkcji e^{-x^2} na zakresie [-5,5] (oczekiwany wyniki: $\sqrt{\pi}$). Całkowaną funkcję zdefiniuj bez nazywania jej, używając wyrażenia lambda.

Wskazówki: podstawowa składnia wyrażeń lambda

Anonimowa funkcja przyjmująca jeden argument typu double:

```
[](double x) { return x; };
```

Opracowanie: Piotr Dziekan.