

## ДИНАМІЧНА БІБЛІОТЕКА ДЛЯ ОБЧИСЛЕНЬ З ФІЗИКИ

<sup>1</sup> Державний торговельно-економічний університет;

<sup>2</sup> Глухівський МЦПО Глухівської міської ради Сумської області

### **Анотація**

*Запропоновано структуру та зміст динамічної бібліотеки для здійснення обчислень з фізики. Розроблена бібліотека надає можливість створення комп'ютерних моделей фізичних явищ з механіки, термодинаміки, електрики і оптики.*

**Ключові слова:** динамічна бібліотека, обчислення, фізика

### **Abstract**

*The structure and content of a dynamic library for performing physics calculations is proposed. The developed library provides an opportunity to create computer models of physical phenomena from mechanics, thermodynamics, electricity and optics.*

**Keywords:** dynamic library, calculations, physics

### **Вступ**

У результаті розвитку інформаційного суспільства значно зросли вимоги до викладання курсу фізики у школі. З іншого боку, глобальна пандемія коронавірусу і війна змусили нас перейти до дистанційної форми навчання. Виявилось, що проводити досліди у дистанційному форматі неможливо. Учень повинен або сам проводити дослід, або використовувати сайт іноземного навчального закладу, де ці моделі є. Доступ до них можуть у майбутньому зробити платним або закрити взагалі.

Саме тому важливим є розроблення власних програмних засобів, які моделюватимуть фізичні процеси і явища, за допомогою яких можна буде провести дослід. Одним із компонентів такого програмного засобу є модуль для здійснення обчислень. У нашому випадку – це динамічна бібліотека функцій, які будуть використовуватися під час обчислень з фізики.

**Мета дослідження** – обґрунтувати зміст динамічної бібліотеки функцій для фізичних обчислень.

### **Результати дослідження**

Функція – це блок коду, який виконує певне завдання і може бути використаний як окремий компонент програми. Функції використовуються для того, щоб уникнути дублювання коду і зробити зміст програми більш зрозумілим.

Бібліотека функцій – це колекція підпрограм, які використовуються для виконання різноманітних завдань. Динамічна бібліотека – це файл, який містить функції та інші ресурси, які можуть бути завантажені під час виконання програми. Динамічна бібліотека (DLL) – це, зазвичай, скомпільований файл, який підключається до програми під час створення і функції з якого використовуються у програмі.

Для створення динамічних бібліотек використовуються мови програмування: C++, C#, Python, Java, Rust. Ці мови відрізняються синтаксисом, функціоналом і можливостями.

Для створення нашої DLL-бібліотеки вибрано мову C#. Основними принципами побудови даної бібліотеки є:

- 1) використання об'єктно-орієнтованої парадигми програмування – тобто, всі функції мають належати певному класу фізичних явищ, тобто являти собою методи;
- 2) компонування функцій у класи здійснюється відповідно до розділів фізики, які вивчаються. Тобто, окремі класи описують функції для обчислень з механіки, термодинаміки, електрики тощо;
- 3) більшість створених функцій – статичні, тобто такі, які викликаються за іменем класу і не потребують створення екземплярів класу.

Динамічна бібліотека складається з таких класів (табл.1). Кожен клас об'єднує методи для обчислення фізичних величин з певного розділу фізики.

Таблиця 1

Класи бібліотеки	
Назва класу	Характеристика
mechanics	Механічні явища
thermo	Явища термодинаміки і молекулярної фізики
electric	Явища з електрики і магнетизму
optic	Оптичні явища

Розглянемо методи на приклад методів одного з класів (табл.2).

Таблиця 2

## Методи класу mechanics

Назва метода	Характеристика
public static double getV(double S,double T)	Отримання швидкості знаючи час і відстань при рівномірному русі
public static double getS(double V,double T)	Отримання відстані залежно від часу та швидкості при рівномірному русі
public static double getT(double V,double S)	Отримання часу залежно від швидкості та відстані при рівномірному русі
public static double getT(double V,double V0,double A)	Отримання часу витраченого на збільшення швидкості при рівномірному прискоренні
public static double getS(double V0,double V,double A)	Отримання пройденої відстані знаючи початкову та кінцеву швидкість й прискорення
public static double getV0(double V,double a,double T)	Отримання початкової швидкості знаючи прискорення, кінцеву швидкість та час
public static double getV(double V0,double a,double T)	Отримання кінцевої швидкості, знаючи початкову, час та прискорення
public static double getA(double V0,double V,double T)	Отримання прискорення з початкової та кінцевої швидкості й витраченого часу
public static double gettimefallingobject(double h)	Знайти час падіння предмета без врахування супротиву навколишнього середовища
public static double getsofffallingobject(double v0,double v)	Знайти відстань яку падав об'єкт за певний проміжок часу(залежно від збільшення швидкості протягом цього проміжку)
public static double getvfromstartoffallingobject(double t)	Знайти швидкість падіння за певний час після початку падіння
public static double getimeoflift(double v0)	Знайти час підйому
public static double gethoflift(double v0)	Знайти відстань на яку піднявся об'єкт
public static double periodob(double t,double n)	Отримати період обертань з часу й кількості обертів
public static double chastotaob(double n,double t)	Отримати частоту обертань з часу та кількості обертів
public static double chastotaob(double T)	Отримання частоти обертів з періоду
public static double periodob(double n)	Знайти період обертів знаючи їх частоту
public static double tzperiodu(double n,double T)	Отримання часу знаючи кількість обертів і період обертань
public static double tzchastoti(double N,double n)	Отримання часу з частоти обертань і кількості обертів
public static double nzperiodu(double T,double t)	Кількість обертів знаючи період і час
public static double nzchastoti(double t,double n)	Кількість обертів знаючи частоту і час
public static double tzblizenia(double v1,double v2,double s)	Знайти час зіткнення знаючи швидкість обох об'єктів та відстань
public static double sviddaleny(double v1,double v2,double t)	Знайти відстань між об'єктами знаючи їх швидкості й час

public static double getrho(double v,double m)	Отримати густину з маси і об'єму
public static double getm(double ro,double v)	Знайти масу з густини та об'єму
public static double getv(double ro,double m)	Знайти об'єм з густини та маси
public static double getfpruz(double k,double x)	Знайти силу пружності з коефіцієнту пружності й видовження
public static double getvid(double l1,double l2)	Знайти видовження
public static double getvidfromfpruz(double f,double k)	Знайти видовження з пружності та сили пружності
public static double getpruzfromf(double x,double f)	Знайти коефіцієнт пружності з сили пружності та видовження
public static double getftyazh(double m)	Знайти силу тяжіння
public static double getmfromtyazh(double f)	Знайти масу з сили тяжіння
public static double getftertya(double u,double n)	Знайти силу тертя з коефіцієнту тертя та сили реакції опори
public static double getufromtertya(double f,double n)	Знайти коефіцієнт тертя
public static double getnfromfertya(double u,double f)	Знайти силу реакції опори
public static double gettisk(double f,double s){}	Знайти тиск
public static double getffromtisk(double s,double p)	Знайти силу з тиску
public static double getsfromtisk(double f,double p)	Знайти площу, знаючи тиск та силу
public static double gethydrostatt(double ro,double h)	Знайти гідростатичний тиск
public static double getrhofromhydrostatt(double ht,double h){}	Знайти густину з гідростатичного тиску
public static double gethfromhydrostatt(double ht,double ro)	Знайти висоту з гідростатичного тиску
public static double getarhf(double ro,double v)	Знайти архімедову силу
public static double getA(double N,double t)	Знайти роботу
public static double getpotuz(double A,double t)	Знайти потужність
public static double gettfromA(double a,double n)	Знайти час роботи
public static double getpotentialenergypidntila(double m,double h)	Знайти потенціальну енергію тіла піднятого над землею
public static double getpotentialenergydeformpruzh(double k,double x)	Знайти потенціальну енергію деформованої пружини
public static double getkinetenergy(double v,double m)	Знайти кінетичну енергію
public static double getvfromkinetenergy(double m,double e)	Знайти об'єм з кінетичної енергії
public static double getmfromkinetenergy(double e,double v)	Знайти масу тіла з кінетичної енергії

public static double getkkd(double q,double a)	Знайти ККД
public static double getafromkkd(double q,double kkd)	Знайти корисну роботу з ККД
public static double getqfromkkd(double a,double kkd)	Знайти затрачену роботу з ККД

Тестування створеної бібліотеки. Для тестування створеної бібліотеки було створено додаток у вигляді екранної форми (рис.1).

Потрібний для тестування клас вибирається за допомогою елемента керування ComboBox1, а потрібний для тестування метод вибирається за допомогою іншого елемента керування ComboBox2. Після цього змінюють написи біля текстових полів, а натискання на кнопку викликає потрібний метод для обчислень.

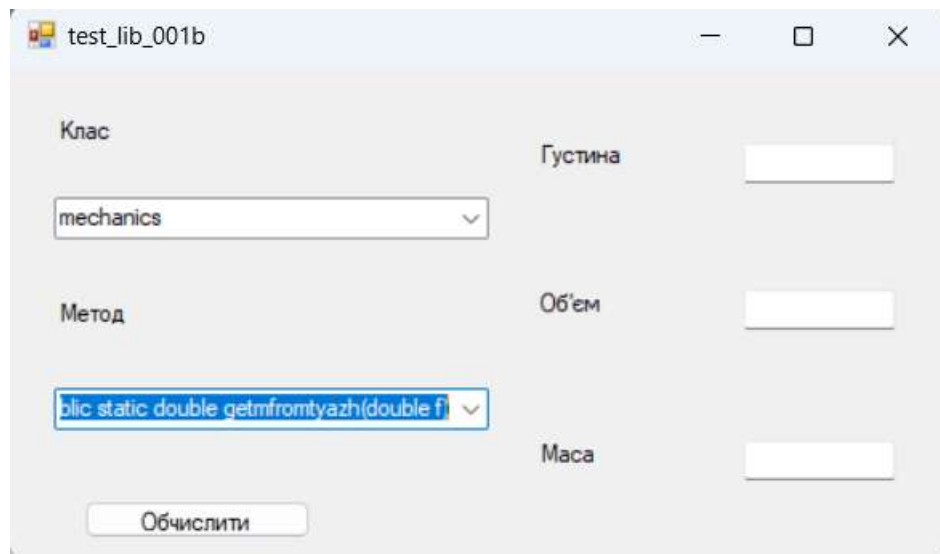


Рис.1. Інтерфейс додатка для тестування бібліотеки

У результаті тестування було перевірено правильність обчислень, які здійснюються всіма функціями бібліотеки.

Призначення створеної динамічної бібліотеки:

- 1) розміщення методів для обчислення фізичних величин з основних розділів фізики;
- 2) підключення цієї бібліотеки до створюваних у майбутньому програм з моделювання фізичних дослідів.

### Висновки

Створена DLL-бібліотека призначена для фізичних обчислень у програмах з моделювання фізичних явищ, які будуть створені у майбутньому. Всі її методи правильно обчислюють фізичні величини. Бібліотека містить статичні методи, тому під час розроблення комп'ютерних моделей програмісту не потрібно буде створювати екземпляри класів.

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. PhET Interactive Simulations [electronic resource]. Access mode: <https://phet.colorado.edu/en/simulations/filter?subjects=physics&type=html.prototype>
2. Німчук Н.І. Комп'ютерне моделювання процесів та явищ на уроках фізики // Збірник наукових праць молодих вчених Кам'янець-Подільського національного університету імені Івана Огієнка. — Кам'янець-Подільський : Кам'янець-Подільський національний університет імені Івана Огієнка, 2019. — Випуск 10. — С.88-89.

**Базурін Віталій Миколайович** – канд.пед.наук, доцент кафедри комп'ютерних наук та інформаційних систем, Державний торговельно-економічний університет, Київ, e-mail: vbazurin@gmail.com

**Базуріна Софія Віталіївна** – вихованка гуртка «Інформатика» Глухівського міського центру позашкільної освіти Глухівської міської ради, Глухів

**Боков Максим Юрійович** – вихованець гуртка «Інформатика» Глухівського міського центру позашкільної освіти Глухівської міської ради, Глухів

**Bazurin Vitalii M.** – candidate of pedagogic sciences, associate professor of the Department of Computer Sciences and Information Systems, State University of Trade and Economics, Kyiv, e-mail: vbazurin@gmail.com

**Bazurina Sofie V.** - pupil of the "Informatics" group of the Hlukhiv city center of extra-curricular education of the Hlukhiv city council, Hlukhiv

**Bokov Maksym Yu.** - pupil of the "Informatics" group of the Hlukhiv city center of extracurricular education of the Hlukhiv city council, Hlukhiv