

ПРОГРАМНА РЕАЛІЗАЦІЯ ОБРОБКИ ВИНЯТКОВИХ СИТУАЦІЙ ПРИ ПРОВЕДЕННІ АНАЛІТИЧНИХ ПЕРЕТВОРЕНЬ ТА ОБЧИСЛЮВАНЬ В ССКА КІДИМ

Головня О.О., Андрєєв Ю.М.

*Національний технічний університет
«Харківський політехнічний інститут», м. Харків*

Програмний комплекс КіДиМ – це спеціальна система комп'ютерної алгебри для проведення розрахунків кінематики і динаміки складних багатомасових систем. вихідні дані для неї готуються в аналітичному (формульному виді). Після зчитування даних програма формує в пам'яті комп'ютера списки констант, змінних з числовими та формульними значеннями, елементів силової, інерційної, пружної та дисипативної структури моделі, інструкцій до розрахунків та операторів, що містять додаткову інформацію до роботи препроцесорів. Попутно с цим ведеться діагностика отриманих даних. Коли ці етапи будуть безпомилково пройдені, ССКА КіДиМ формує в пам'яті ПК відповідні динамічні, кінематичні, або статичні рівняння. Далі в залежності від дій користувача можуть бути проведені розрахунки, що відповідають поставленому завданню, – отримані потрібні аналітичні викладки, побудовані графіки, табличні дані, представлені анімації процесів.

Природно, що в таких даних можуть бути синтаксичні та семантичні помилки. Також можуть бути ситуації, коли в процесі моделювання кінематичного або динамічного процесу деякі формули вироджуються. Такі помилки заздалегідь приводять до створення виняткових ситуацій, результатом яких в кращому випадку буде закінчення роботи всієї програми без розкриваючих ситуацію повідомлень, або з мінімальними повідомленнями, що мало допомагають. Це пояснюється тим, що розрахунок формул – це суттєво рекурсивний процес і поява виняткової ситуації на якомусь її рівні приводить до дуже важкої задачі розкрутити рекурсії назад і отримати повну картину того, що трапилося.

Для розв'язання таких питань в даній роботі використано механізм генерації і обробці виняткових ситуацій, що мається в мові програмування C++. Він полягає в тому, що ті фрагменти коду, в яких використовуються такі рекурсивні виклики програм, перш за все, там, де організовано розрахунки в циклах, обрамляються блоком **try**{}, за яким ставиться блок **catch(){...}**, якому параметром передається покажчик на екземпляр спеціально створеного класу, що інкапсулює всю необхідну для розкриття ситуації інформацію. Ця інформація (момент часу процесу, клас помилки, формула, яка розраховувалася, оброблялася, перетворювалася) формується в точці виникнення і доповнюється в блоці **catch**.

В докладі демонструються приклади роботи при діагностиці зациклювання формул, складних синтаксичних конструкцій, виняткових математичних ситуацій при розрахунках. таке рішення суттєво підвищує якість всього програмного комплексу.