

Учредитель:
КрТО МАКНС

14'04

Академический вестник
Криворожского территориального
отделения Международной Академии
компьютерных наук и систем
(КрТО МАКНС)

Редакционная коллегия

Назаренко В.М. ,

академик-секретарь отделения, д.т.н., проф., - главный редактор

Ефименко Л.И. ,

к.т.н., доцент КТУ, - ответственный редактор

Марусич Ю.Ю. , -

технический редактор

Члены редколлегии:

Шапурин А.В. , академик МАКНС проф., докт.техн.наук

Азарян А.А. , эксперт МАКНС, проф., докт.техн.наук

Толмачев С.Т. , эксперт МАКНС, проф., докт.техн.наук

Евтехов В.Д. , академик МАКНС, проф., докт.г-м.наук

Соловьев В.М. , эксперт МАКНС, проф., докт.ф-м.наук

Учитель А.Д. , эксперт МАКНС, проф., докт.техн.наук

Садовой А.В. , эксперт МАКНС, проф., докт.техн.наук

Щупов В.П. , проф., докт.техн.наук

Бережной Н.Н. , эксперт МАКНС, проф., докт.техн.наук

Губин Г.В. , эксперт МАКНС, проф., докт.техн.наук, академик АГН
Украины

Рудь Ю.С. , проф., докт.техн.наук

Трегубов В.А. , эксперт МАКНС, проф., докт.техн.наук, член-
кор.АГН Украины

Каварма И.И. , проф., докт.техн.наук, член-кор.АГН Украины

Осадчук Ю.Г. , канд.техн.наук

Журнал зарегистрирован
Министерством информации Украины
Регистрационный номер № 3020
от 26.01.1998 г.

Издается на украинском и русском языках.
Печатается по решению Ученого Совета
Криворожского технического университета и
бюро КрТО МАКНС

Адрес редакции
50027, г.Кривой Рог,
ул.ХХІІ партсъезда, 11

Тел. (0564) 74-14-35
71-93-87
71-93-83
Факс 29-19-91

Издатель :
КрТО МАКНС

Директор издательства
Назаренко М.В.

Выпускающий редактор
Марусич Ю.Ю.

Художественное оформление и
компьютерная верстка
Марусич И.В.

Колонка редакции

Редакция предлагает за-
интересованным лицам и организациям
присылать научные и рекламные
материалы для публикации в нашем
журнале.

Экспертная коллегия по рецензированию научных статей.

- Назаренко В.М.** - проф., докт. техн. наук, зав. кафедрой информатики, автоматизации и систем управления Криворожского технического университета (КТУ), академик МАКНС .
- Учитель А.Д.** - проф., докт. техн. наук, зав. кафедрой электромеханического оборудования металлургических заводов Государственной металлургической академии Украины, эксперт МАКНС
- Евтехов В.Д.** - проф., докт. геол.-минер. наук, зав. кафедрой минералогии КТУ , академик МАКНС .
- Шапурич А.В.** - проф., докт. техн. наук, академик МАКНС .
- Ткачев В.В.** - проф., докт. техн. наук, зав. кафедрой автоматизации производственных процессов Национальной горной академии Украины.
- Марюта А.Н.** - проф., докт. техн. наук, зав. кафедрой АСУ и информатики Днепропетровского государственного университета.
- Хорольский В.П.** - проф., докт. техн. наук, зав. кафедрой менеджмента Криворожского экономического института Национального экономического университета.
- Качан Ю.Г.** - проф., докт. техн. наук, ген. директор Межрегионального учебного центра Энергофахсервис.
- Качура Е.В.** - проф., докт. техн. наук.

СОДЕРЖАНИЕ

АВТОМАТИЗАЦІЯ ВЕЛИКИХ СИСТЕМ, ПРОГРЕСИВНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА ПРОБЛЕМИ ШТУЧНОГО ІНТЕЛЕКТУ В ГІРНИЧІЙ СПРАВІ

<i>Зубов Д.А., Ульшин В.А.</i> Оценка адекватности двухканальной нелинейной модели технологических процессов углеобогащения (на примере флотации)	8
<i>Лосихин Д.А., Тришкин В.Я.</i> Оценка эффективности систем управления на стадии проектирования	13
<i>Варфоломеева И.В.</i> Квантовый подход инженерии знаний для принятия прогнозных и идентификационных решений	14
<i>Борин В.С.</i> Застосування фазі-логіки в автоматизованій системі управління процесом абсорбційної осушки природного газу на компресорних станціях магістральних газопроводів	18
<i>Купін А.І., Гончаров Є.В.</i> Структура інформаційної підсистеми АСУ ІнГЗК на основі комп'ютерної мережі	20
<i>Никитин А.И., Купин А.И.</i> Нейронные сети как новый подход к управлению технологическим оборудованием	23
<i>Щокін В.П., Щокіна О.В.</i> Формалізація функцій приналежності нечіткої нейромережевої моделі ймовірносної оцінки ефективної реалізації проекту	26
<i>Воловик В.П., Корсун В.І.</i> Застосування стохастичних мереж при плануванні гірничих робіт	28
<i>Внукова Т.И.</i> Измерительно-вычислительный комплекс для измерения поверхностей деталей сложной формы	29
<i>Коваленко І.В., Корсун В.І.</i> Структурно-автоматне моделювання гірничо-транспортної системи кар'єру	32
<i>Дронь Н.М., Гринчишин Ю.Л., Хорольский П.Г.</i> Концепция автоматизации проектирования сложных ракетно-космических систем	33
<i>Савицький О.І., Акіменко С.О., Нікітін А.І.</i> Особливості застосування SCADA-систем для диспетчеризації гірничих процесів	40
<i>Льченко В.О.</i> Оптимізація роботи в системі „1С: Підприємство” на базі компоненти «Windows terminal server»	44
<i>Тимченко А.А., Махинько Н.В.</i> Системное моделирование потоков в технических установках	44

<i>Тимченко А.А.</i> Самонастраивающиеся системы управления движением с аналитическими нелинейностями	46
<i>Фокин А.Г., Кисловский Н.И.</i> Экспертная система автоматизированного проектирования технологий	46
<i>Фокин А.Г.</i> Использование таблиц решений в сложных информационных системах	51
<i>Барановський С.С., Лобов В.Й.</i> Модульні алгоритми і робочі програми для побудови автоматизованих систем по обліку товарі на складах	57
<i>Хоменко С.А., Белкін Д.А.</i> Автоматизоване управління персоналом в умовах сучасних підприємств	60
<i>Волкова Н.В.</i> Інформаційні системи для визначення норм витрат матеріальних ресурсів та їх зберігання	64

НОВІ ПІДХОДИ В НАВЧАННІ ТА ВИХОВНОМУ ПРОЦЕСІ

<i>Числова Є.А.</i> Організація навчального процесу на основі системного підходу	68
<i>Завізна Н.С.</i> Педагогічний аспект індивідуалізації навчального процесу на основі застосування комп'ютерів у вищій педагогічній школі	69
<i>Внуков И.П., Зянчурина И.Н.</i> Виртуальный лабораторный практикум по курсу «Компьютерные системы управления технологическими процессами»	72
<i>Белкіна С.Д.</i> Удосконалення заочної форми навчання шляхом втілення елементів дистанційної освіти	77
<i>Братков С.М.</i> Технократический подход к системе образования	78
<i>Маслова Н.В.</i> Гуманитация и гуманитаризация в современном образовательном процессе технического ВУЗа	79
<i>Волик Б.А.</i> Использование элементов виртуального моделирования в учебном процессе	82
<i>Бойко С.М.</i> Методичні рекомендації для проведення робот из адаптації студентів нового набору	84
<i>Бантос М.М., Фалько Л.В.</i> Использование элементов методологии соционики в учебно-воспитательном процессе ВУЗа	86
<i>Туравинина О.Н., Чубаров В.А.</i> Особенности проведения занятий по дисциплине «Информатика и компьютерная техника» в финансовых ВУЗах	88

<i>Боско О.М., Гринь Н.В.</i> Залучення представників корпорацій-виробників програмних продуктів до викладання дисциплін циклу "Інформаційні системи" у вищих навчальних закладах	91
<i>Фалько Л.В.</i> Новые аспекты применения классических принципов педагогики и соционики для повышения качества обучения в высшей школе	92
<i>Бобилев Д.Є.</i> Спецкурс „Метод граничних елементів у задачах геомеханіки“ (для гірничих спеціальностей ВУЗів) та методика його викладання	95
<i>Бобилева В.О.</i> Використання інформаційних технологій в процесі управління формуванням структури капіталу підприємства	97
<i>Настенко І.В.</i> Особливості використання проблемних методів навчання в контексті педагогічної взаємодії викладача і студентів в процесі викладання інформатики	99
<i>Конченко Л.Л.</i> Причини неуспішності студентів-першокурсників і шляхи їх подолання	102
<i>Полищук А.П., Семериков С.А.</i> Использование средств объектно-ориентированного программирования для компьютерной реализации векторной, матричной и полиномиальной алгебр	105
<i>Семериков С.О.</i> Принципи застосування об'єктного підходу до розробки математичного програмного забезпечення	110
<i>Теплицкий І.А.</i> Информационная культура и информационная безопасность как факторы выживания в информационном обществе	115
<i>Денисюк В.А., Семериков С.О., Теплицкий І.О.</i> Методичні основи дистанційного тестування знань засобами FTN-технологій	120
<i>Леонова Н.А., Моисеенко Н.В., Семериков С.А.</i> Пропедевтика метода наименьших квадратов в курсе «Компьютерные технологии в научных исследованиях»	125
<i>Олейникова Т.Ю., Данченко Е.Б.</i> Задача формування структури модулів дисциплін при модульно-рейтинговій системі навчання	129
<i>Гуливец А.А.</i> К вопросу момента силы относительно оси	131

ПРОБЛЕМИ ОХОРОНИ ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ В СФЕРІ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

<i>Назаренко В.М., Назаренко М.В., Смирнова Н.В.</i> Програмне забезпечення: кумулятивна охорона	134
--	-----

<i>Галенко В.П.</i> Ескізний проект реконструкції Міжнародної патентної класифікації	137
<i>Лантева И.С.</i> Стратегия патентной охраны инженерной разработки	141
<i>Соловьев В.В.</i> Возможность использование языка моделирования UML для моделирования законодательства по интеллектуальной собственности	144
<i>Кошулько Г.М., Смирнова Н.В.</i> Використання інноваційних методів у викладанні дисципліни „Інтелектуальна власність в інформаційних технологіях” в умовах дистанційного навчання	146
<i>Конченко Л.Л.</i> Необхідність та стан викладання дисциплін з інтелектуальної власності студентам спеціальностей інформаційного профілю КТУ	148
<i>Зайцева А.Д., Фурманова Н.В., Чухарев С.М.</i> Коммерциализация объектов интеллектуальной собственности	150

Організаційно таке навчання може бути побудоване тільки у формі самостійної аудиторної роботи студентів над матеріалом навчально-методичних посібників при своєчасній консультативній допомозі викладача. Крім того, має бути належним чином організований контроль і перевірка знань студентів. Вивчення кожного розділу навчального матеріалу повинно закінчуватись складанням студентом звітів, в яких він показує розуміння взаємозв'язку основ теорії і типових прийомів та методів діяльності. Одним із завдань для викладача під час проведення контролю є надання студентам пояснень щодо отриманої оцінки, допущених помилок та рекомендацій для покращення результатів наступних контрольних-залікових заходів.

Найкраще такий вид навчання застосовувати у рамках модульно-рейтингової системи, яка вимагає розподілу дисципліни на окремі частини (модулі), послідовного вивчення студентами кожного модуля, проведення після кожного модуля контрольних-залікових заходів (найкраще у вигляді тестів). Всі досягнення студентів оцінюються умовними заліковими одиницями, які в кінці семестру переводяться в оцінку за п'ятибальною шкалою.

Хочеться наголосити, що самостійна робота студентів не означає, що вони мають засвоювати зміст дисципліни лише хай із найсучасніших, але методичних посібників. Головною задачею викладача було і залишається донесення до кожного слухача навчального матеріалу. Річ у тім, що студенти ще з початку вивчення дисципліни мають на руках тексти всіх лекцій, і йдуть на заняття вже попередньо ознайомившись з конкретною темою. Безпосе-

редньо на парі викладач пояснює матеріал, наводить приклади, демонструє досліди, відповідає на запитання. Студентам вже не потрібно нічого записувати: вони лише слухають пояснення і за бажанням занотують цікаві моменти. Це значно звільняє час, бо викладач встигає пояснити більше матеріалу, ніж у випадку задиктовування лекції. Крім того, студенти не виконують механічну роботу, а саме осмислюють матеріал, що сприяє кращому його засвоєнню.

Така методика вже декілька років застосовується викладачами кафедри Інформатики, автоматики і систем управління Криворізького технічного університету при вивченні технічних дисциплін теоретичного характеру. І можна констатувати, що за умови своєчасного проведення поточних контрольних заходів, вона дає стійкі позитивні результати: студенти краще вчаться, у них підвищується пізнавальний інтерес, з'являється мотивація навчання, зацікавленість предметом, самостійність, впевненість в своїх силах. Крім того, заняття завжди проходять нешаблонно, цікаво, що психологічно змушує "невстигаючих" підтягуватись до рівня середнього студента.

Підводячи підсумки, хочеться зазначити, що є велика кількість негативних чинників, які впливають на успішність студентів-першокурсників, та й взагалі всіх студентів. Задача викладачів – своєчасно помітити відставання, надати посильну допомогу та щоденно працювати над собою, щоб зробити свої заняття ще цікавішими та привабливішими. Хай щастить викладачам у цій нелегкій справі, як і студентам, що колись, можливо, згадають нас добрим словом за ті знання, що ми вклали в їх голови, та думки, які їм нав'яли.

УДК 519.88

©Полищук А.П., Семериков С.А., 2004

Использование средств объектно-ориентированного программирования для компьютерной реализации векторной, матричной и полиномиальной алгебр

Полищук А.П., канд. техн. наук, доц., Семериков С.А., канд. пед. наук, доц. (КГПУ, г.Кривой Рог)

В статье описывается созданная авторами объектно-ориентированная математическая библиотека, используемая при чтении курсов «Методы машинных вычислений» и «Объектно-ориентированное программирование» в Криворожском государственном педагогическом университете.

Задачи, ради которых и были созданы компьютеры – рутинные расчёты производственного, научного и военного характера, – потребовали создания целого класса новых методов, ориентированных не на

ручные, а на машинные вычисления. Первые языки программирования не обладали удобными средствами для отражения таких часто используемых в вычислительной математике объектов, как матрицы, вектора,

полиномы и т.д. Дальнейшее развитие языков программирования шло по пути встраивания математических объектов в языки как типов данных, что вело к их усложнению (рис.1). Так, например, попытка сделать универсальный язык Ада, в котором

есть даже такие типы данных, как словари и очереди, привела к тому, что количество ключевых слов в нём превысило 350, сделав его практически непригодным для изучения и использования.

Языки общего назначения
(Си, Паскаль, Бейсик)

- Отсутствие векторных, матричных и полиномиальных типов данных;
- Простота изучения.

Проблемно-ориентированные языки
(Ада, Лисп, Кобол)

- Наличие множества специализированных типов данных, в том числе матричных и полиномиальных;
- Сложность в изучении и использовании.

Объектно-ориентированные языки
(Objective C, C++, Object Pascal)

- Возможность создания любых требуемых типов данных;
- Относительная простота в изучении и удобство в использовании.

Рис.1. Классификация языков программирования (согласно [3])

Компромиссным решением между этими двумя крайностями может быть следующее: пусть программист сам создаёт типы данных, которые ему необходимы в его профессиональной деятельности. Языки программирования, в которых реализован такой подход, называют объектно-ориентированными. Это, с одной стороны, позволяет сделать язык достаточно лёгким путём уменьшения количества ключевых слов, а с другой – расширяемым, приспособляемым к конкретным задачам введением ключевых слов для создания и использования новых типов данных.

Язык C++, изучающийся нами в курсе информатики, является объектно-ориентированным языком. Типы данных, которые мы поставили своей целью создать средствами этого языка, определились из специфики подготовки учителей математики [4, 5].

Одной из самых важных учебно-методических линий в курсе алгебры средней школы является линия уравнений и неравенств, а также их систем. В высшей школе эта линия продолжается на более высоком уровне в курсе линейной алгебры и

векторных пространств, что позволило нам выделить следующие математические объекты: вектора, матрицы и многочлены. Наш выбор обуславливался ещё тем, что все они тесно связаны друг с другом. Так, например, матрицу можно рассматривать как упорядоченный кортеж арифметических векторов, а многочлен можно задать вектором его коэффициентов.

Для каждого из этих типов в процессе выполнения объектно-ориентированного анализа понадобилось определить множество процедур, в большинстве своём – бинарных алгебраических операций над элементами соответствующих множеств. Результатом работы явилось создание библиотеки классов параметризованных векторных, матричных и полиномиальных объектов, расширяющих возможности языка C++ по работе с такими объектами [1]. Рассмотрим детальнее механизм реализации этой библиотеки.

Базовым в нашей иерархии классов является класс для работы с арифметическими векторами (рис.2а). В нашей интерпретации вектор – это упорядоченный кортеж некоторых объектов (или, иными сло-

вами, массив определённой длины). Тип числа, составляющих компоненты вектора, не является строго фиксированным – этот тип можно использовать как параметр конструктора векторного объекта, то есть вы можете иметь целочисленные, вещественные, комплексные и т.д. вектора различных длин. Основные операции, определённые для векторов – это сложение, вычитание,

отрицание, скалярное умножение, умножение вектора на скаляр, сравнение векторов, вывод вектора в поток и ввод его из потока, нахождение модуля вектора и нормирование его по модулю, а также ряд операций сокращённого сложения, вычитания и т.д. Разумеется, большинство этих операций определены только для векторов совпадающих размерностей.

Структура класса Vector

Данные

- n* - размерность вектора (натуральное число)
- vec* - указатель на компоненты вектора (координаты вектора в *n*-мерном пространстве)

Методы

1. Конструкторы (по умолчанию, копирования, из файла, из массива)
2. Деструктор
3. Операция сложения
4. Операция сокращённого сложения
5. Операция вычитания
6. Операция сокращённого вычитания
7. Скалярное произведение двух векторов
8. Умножение скаляра на вектор (вектора на скаляр)
9. Сокращённое умножение вектора на скаляр
10. Вывод вектора в поток
11. Ввод вектора из потока
12. Операция присвоения
13. Унарный минус
14. Модуль вектора
15. Нормирование вектора по модулю
16. Проверка векторов на равенство и неравенство
17. Индексирование элементов вектора
18. Метод для определения размерности вектора

а

Структура класса Polynom

Данные *унаследованы из класса Vector*

Методы

1. Конструкторы (по умолчанию, копирования, из файла, из массива)
2. Унарный минус
3. Операция сложения
4. Операция сокращённого сложения
5. Операция вычитания
6. Операция сокращённого вычитания
7. Умножение полинома на полином
8. Умножение скаляра на полином (полинома на скаляр)
9. Сокращённое умножение полинома на скаляр
10. Сравнение полиномов
11. Вычисление значения полинома в заданной точке
12. Деление с остатком, целая часть и остаток от деления
13. Присвоение
14. Производная заданного порядка
15. Интеграл заданной кратности
16. Степень полинома
17. Сокращённая степень полинома
18. Получение коэффициента при заданной степени
19. Приведение полинома к каноническому виду

б

Рис.2. Структура векторного (а) и полиномиального (б) классов

Для реализации арифметических операций был использован механизм перегрузки операций, благодаря которому, например, при сложении двух векторов в программе достаточно поставить знак «+» между объектами векторного класса точно так же, как это делается при сложении стандартных целых и вещественных чисел. Такая запись является более естественной, чем вызов функции `Add`, хотя, по сути, ничем от неё не отличается. Это свойство является особенностью именно C++, в отличие от других объектно-ориентированных языков.

Следующим классом, разработанным нами, был класс, предназначенный для хранения и использования, пожалуй, самых важных алгебраических объектов – матриц. Как видно из рис.3 (верхняя часть), структура матричного класса подобна структуре

векторного, но более содержательная. Это обусловлено множеством операций, которые необходимо было запрограммировать для эффективной работы с новым типом данных – «матрица». Как и векторный класс, матричный богат конструкторами, которые позволяют создавать матрицы, инициализируя их данными из памяти, из файла, из другой матрицы и так далее. Кроме того, допустимо создание «пустой», т.е. нулевой матрицы заданного размера.

В качестве внутреннего представления матрицы был выбран упорядоченный кортеж векторов – объектов векторного класса, рассмотренного нами ранее, что позволяет использовать для матричных операций перегруженные операции класса «вектор». Например, для сложения двух матриц совпадающих размерностей нам достаточно сложить вектора, их составляющие, что по-

звolyает вместо двух циклов сложения элементов матриц использовать один цикл сложения векторов, составляющих строки матрицы.

Для матриц совпадающей размерности определены операции сложения, вычитания, а также их сокращённые аналоги; умножение матрицы на скаляр, транспонирование, сравнение матриц, вывод в поток и ввод из потока. Для отдельных типов матриц определена операция умножения.

Практический интерес при работе с матрицами представляет решение систем линейных алгебраических уравнений, задаваемых соответственно матрицами коэффициентов при неизвестных и вектор-столбцом свободных членов. В связи с этим были реализованы следующие методы решения СЛАУ: метод Гаусса с выбором главного элемента, метод ортогонализации векторов матрицы, метод обратной матрицы, метод Крамера.

Структура класса Matrix

Данные

m, n - размерность матрицы (количество строк и столбцов)

mtx - указатель на векторы-компоненты матрицы

Методы

1. Конструкторы (по умолчанию, копирования, из файла, из массива)
2. Деструктор
3. Операция сложения
4. Операция вычитания
5. Операция умножения
6. Умножение матрицы на скаляр (скаляра на матрицу)
7. Вывод матрицы в поток и ввод матрицы из потока
8. Метод ортогонализации
9. Метод Гаусса
10. Вычисление детерминанта
11. Операция присвоения
12. Операции сокращённого умножения, сложения и вычитания
13. Степень матрицы
14. Транспонирование матрицы
15. Обращение матрицы
16. Проверка матриц на равенство и неравенство
17. Индексирование элементов матрицы
18. Методы для определения размерности матрицы

```
Фрагмент программы, решающей СЛАУ

matrix a(n,n), b(n,1);
...
matrix x=row(a,-1)*b;
```

```
Фрагмент программы восстановления
линейных зависимостей по МНК

matrix x(count,n), y(count,1);
...
matrix a=row(x*x,-1)*(x*y);
```

Рис. 3. Структура матричного класса

Вычисление обратной матрицы и детерминанта можно производить либо аналитически, либо численно. Нами были реализованы оба этих подхода. Так, Вы можете воспользоваться рекурсивной функцией вычисления детерминанта разложением его по какой-либо строке и функцией обращения матрицы с использованием алгебраических дополнений. При этом с ростом порядка матрицы количество операций сло-

жения и умножения возрастает настолько, что эти методы становятся малоэффективными. Более эффективным является вычисление детерминанта и обратной матрицы косвенно, путём решения системы уравнений.

Перегрузка перечисленных операций даёт нам возможность записи операций над матрицами наиболее приближенно к алгебраической форме. К примеру, решение системы уравнений может быть записано

как решение матричного уравнения $AX=B$, $X=A^{-1}B$, где операция возведения в степень «-1» не что иное, как перегруженная функция обращения матрицы, а «*» - операция умножения матриц. На рис.3 (нижняя левая часть) показан фрагмент решения этого уравнения в терминах нашего класса.

Наличие, наряду с умножением и обращением, операции транспонирования, позволяет нам одной строчкой программы записать решение задачи МНК – метода наименьших квадратов. Пусть X и Y – соответственно матрицы независимых и зависимых переменных, A – неизвестный вектор оценки МНК. Тогда $A=X$ транспонированное, умноженное на X (всё в минус первой степени), умноженное на произведение транспонированной матрицы независимых переменных на вектор-столбец зависимых переменных (рис.3, нижняя правая часть).

На рис.2б представлена структура класса для работы с многочленами от одной переменной. Как видно из этой схемы, данный класс базируется на векторе – действительно, операции над многочленами сводятся к действиям над коэффициентами при соответствующих степенях неизвестной. Это даёт возможность использовать арифметический вектор для представления многочлена. Как и в предыдущих классах, полиномиальные объекты имеют набор методов для конструирования, сложения, вычитания, умножения полинома на полином и полинома на скаляр, сравнения, деления с остатком и т.п.

Для полиномиальных объектов мы можем найти производную любого порядка и неопределённый интеграл любой кратности в аналитической форме; результатом будет полином, коэффициенты которого получают по соответствующим правилам. Кроме того, в любой точке мы можем найти функциональное значение полинома.

В задачах прикладной математики полиномиальные объекты, как правило, используются при решении алгебраических уравнений, что побудило нас к реализации ряда методов решения уравнений разного порядка. Согласно основной теореме алгебры, полином n -ной степени имеет ровно n корней. Это позволяет считать, что решением полиномиального уравнения в комплексной области является комплексный вектор решений с размерностью, равной степени многочлена, компонентами которого являются его корни.

Для многочленов с действительными коэффициентами нами были рассмотрены метод Кардано-Тарталья для решения урав-

нений 3-ей степени и базирующийся на нём метод Феррари решения уравнений 4-ой степени. Более универсальными методами для комплексных многочленов являются методы решения квадратных уравнений и метод Ньютона для поиска корней многочлена любой степени. В последнем методе мы, находя очередной корень, отделяем его, понижая степень многочлена по схеме Горнера, ищем корень многочлена меньшей степени и т.д., до отделения всех корней.

Последняя часть нашей библиотеки включает в себя алгоритмы и программы решения проблемы собственных значений для несимметричных комплексных матриц. Для этого используются все разработанные нами классы – вектора, матрицы и полиномы, существенно сокращая объём программы и повышая её наглядность.

Основные результаты и выводы.

1. Представленная библиотека свидетельствует о том, что выбранная методика создания математических объектов с использованием средств объектно-ориентированного программирования, наиболее пригодна для компьютерной реализации прикладных задач.
2. Для создания библиотек математических классов целесообразно использовать язык C++, что дает возможность использовать механизм перегрузки операций, а также предоставляет программисту широкий диапазон стандартных математических функций, в том числе для комплексных чисел.
3. Работая с матричными объектами, целесообразно применять объектно-ориентированную технологию; разработанная нами библиотека классов позволяет решать задачи регрессионного анализа за короткое время с малыми трудозатратами.
4. Созданная библиотека может быть использована для расчётов в электротехнике, физике и других областях науки, а также, благодаря доступности её в исходных текстах с комментариями, как учебный материал для курса «Объектно-ориентированное программирование на языке C++» и «Методы машинных вычислений» [2].

Список литературы.

1. Єпішин О.В., Федоренко Д.С., Поліщук О.П. Створення спеціалізованих математичних класів та їх застосування при викладанні чисельних методів // Фізика. Математика. Нові технології

- навчання. Збірник матеріалів Всеукраїнської студентської науково-практичної конференції. – Кіровоград: РВГІЦ КДПУ ім. В. Винниченка. – 1999. – С.36-37.
2. Полищук А.П., Семериков С.А. Методы вычисления в классах языка С++: Учебное пособие. – Кривой Рог: Издательский отдел КГПИ, 1999. – 350с.
 3. Полищук А.П., Семериков С.А., Грищенко Н.В. О выборе языка программирования для начального обучения //Комп'ютерне моделювання та інформаційні технології в природничих науках. – Кривий Ріг: Видавничий відділ КДПУ, 2000. – С. 212-228.
 4. Семериков С.О. Застосування об'єктно-орієнтованого програмування до викладання курсу чисельних методів //Математика, її застосування та викладання: Матеріали міжвузівської регіональної наукової конференції. – Кіровоград: РВГ ІЦ КДПУ ім. В. Винниченка. – 1999. – С.134-137.
 5. Семериков С.О. Чисельні методи: об'єктний підхід //Комп'ютерно-орієнтовані системи навчання. Збірник наукових праць. Випуск 2. – К.: НПУ ім. М.П. Драгоманова, 2000. – С.122-128.

УДК 378.147+518.5

©Семериков С.О., 2004

Принципи застосування об'єктного підходу до розробки математичного програмного забезпечення

*Семериков С.О., канд. пед. наук, доц.
(КДПУ, м.Кривий Ріг)*

В статті викладено основні принципи об'єктного підходу та його реалізація у об'єктно-орієнтованому програмуванні, показана природна спільність математичних та програмних об'єктів, наведено рекомендації щодо розробки математичного програмного забезпечення.

«Наука про те, як замість лише швидкодії машини використовувати чисельні методи та бібліотечні програми, переживає період дитинства і є однією з найважливіших областей дослідження у майбутньому» [1, с.198]. Ці слова Р.В.Хеммінга, сказані майже 40 років тому, і досі не втрачають своєї актуальності. Тенденція до створення та систематичного використання математичних бібліотек, що виникла на початку 60-х рр., наприкінці 80-х рр. стає домінуючою. Проте розвиток чисельних методів зумовлений сьогодні також розвитком засобів обчислювальної техніки та технології програмування. Так, поява паралельних ЕОМ породила новий клас чисельних методів, орієнтованих на паралельні обчислення, а розвиток об'єктно-орієнтованого програмування – новий напрямок: Object-oriented numerics (OON), або об'єктні обчислення, який зараз є провідним напрямком у чисельних методах [2].

Об'єктно-орієнтоване програмування (ООП), що отримало широке розповсюдження як потужна програмна технологія, є у наш час вагомим альтернативним традиційним процедурним методом програмування. Популярність ООП у чималій мірі визначається концептуальною цілісністю та більш сильною формою структуризації про-

грамного забезпечення (ПЗ), що створюється на його основі. Використання ООП прискорює процес розробки програм, даючи при цьому можливість гнучкої та природної модифікації існуючого ПЗ. Найбільш рельєфно можливості ООП проявляються при створенні досить складних програмних продуктів, до яких, зокрема, відносяться проблемно-орієнтовані бібліотеки.

Перш ніж розглядати ООП, доцільніше спочатку розглянути його підґрунтя – об'єктний підхід, який є більш загальною технологією дослідження та пізнання, як це пропонує А.П.Єршов. У своїй роботі «Про об'єктно-орієнтовану взаємодію з ЕОМ» він надає об'єктно-орієнтованому програмуванню більш широкий зміст, ніж програмуванню лише з використанням об'єктно-орієнтованих мов. У якості одного з прикладів об'єктно-орієнтованої взаємодії програмувача з ЕОМ А.П.Єршов посилається на Е-практикум як реальну систему автоматизованого конструювання програм, особливо підкреслюючи при цьому тезу про перспективність та універсальність об'єктно-орієнтованої взаємодії. Отже, розглянемо основні передумови впровадження об'єктного підходу у практику викладання чисельних методів, і почнемо з психологічних передумов.