

УДК 372.851



Г.Ф. Сафонова,

к.т.н., ст. викладач
Одеський національний
політехнічний
університет
e-mail:uskach_83@mail.ru

ПРОВЕДЕННЯ ІНТЕГРОВАНИХ ЗАНЯТЬ З МАТЕМАТИКИ ЯК ОДИН ІЗ СПОСОБІВ РЕАЛІЗАЦІЇ ЗВ'ЯЗКІВ З МАБУТНЬОЮ СПЕЦІАЛЬНІСТЮ СТУДЕНТІВ КОЛЕДЖУ

А.Ф. Сафонова. Проведення інтегрованих занять по математике как один из способов реализации связей с будущей специальностью студентов колледжа. Был рассмотрен вариант интегрированного занятия по математике с элементами основ программирования, как один из способов реализации связей с выбранной специальностью.

A.F. Safonova Leadthrough of computer-integrated employments on mathematics as one of methods of realization of connections with future speciality of collegians. The variant of integrirovannogo employment was considered on mathematics with the elements of bases of programming, as one of methods of realization of connections with the chosen speciality.

Вступ. Проблема вивчення математики у ВНЗ пов'язана в першу чергу з питанням мотивації пізнавальної діяльності студентів, їх зацікавленості предметом. Викладач, спершу, має відповісти на питання студентів, не тільки для чого їм потрібно вивчати розглянуту тему навчальної програми, а і як можна застосувати даний матеріал у їх майбутній спеціальності. Один із способів вирішення цієї проблеми є впровадження в процес навчання інтегрованих занять.

В статі розглядається план-конспект інтегрованого заняття з предмету «Математика» з елементами дисципліни «Основи програмування», розроблений для студентів спеціальностей «Розробка програмного забезпечення» та «Обслуговування комп'ютерних систем та мереж». Заняття відноситься до розділу навчальної програми математики «Елементи комбінаторики» та містить елементи тем дисципліни «Основи програмування» таких, як: «Цикли», «Підпрограми».

Матеріал і результати дослідження. Розглянемо конспект інтегрованого заняття з математики на прикладі:

Тема заняття: «Сполуки без повторень. Побудова трикутника Паскаля з застосуванням засобів мови програмування Pascal».

Мета заняття: навчитися будувати трикутник Паскаля, використовувати у програмуванні підпрограми.

Тип заняття: повторення та засвоєння нових знань.

План заняття:

1. Вступ.
2. Повторення (формула числа сполук та їх властивості).
3. Вид таблиці.
4. Допоміжні алгоритми та підпрограми.
5. Складання програми.
6. Заповнення таблиці.
7. Виявлення ще одної властивості сполук.
8. Підсумки.

1. Вступ.

Сьогодні на занятті ми з вами побудуємо спеціальну таблицю, яка допоможе нам побачити цікаві закономірності та властивості чисел, які входять в неї. Також ви згадаєте такі засоби мови програмування, як підпрограми.

Таблиця, заповненням якої ми займаємося, складається з чисел сполук, які записуються у визначеному порядку.

2. Повторення.

- Що таке сполука?
- Назвіть сполуки та їх кількість, які можна скласти з цифр 1,2,3 по 2 елемента у кожній сполуці? Відповідь: 3 (12,13,23)
- Чому ви не використали число 21, це ж інше число? Відповідь: 21 та 12 – одна і та ж сполука, т. як. різні сполуки повинні відрізнятися хоча б одним елементом.
- За якою формулою можна обчислити число сполук?

Відповідь:
$$C_n^k = \frac{n!}{k!(n-k)!}$$

- Які властивості числа сполук вам відомі? Відповідь: 1) $C_n^n = 1$;
2) $C_n^0 = 1$, 3) $C_n^k = C_n^{n-k}$. Де використовується ця властивість? Відповідь: при обчисленні великого числа сполук; 4) $C_n^1 = n$.

Але це не всі властивості числа сполук.

3. Вид таблиці.

Для знайомства з іншими властивостями числа сполук, ми запишемо числа сполук по рядкам, починаючи з $n=0$ (у першому рядку), $n=1$ у другому рядку, $n=2$ у третьому ... $n=7$ у восьмому у вигляді пірамідки.

Подивіться, які клітинки нашої таблиці ми можемо вже заповнити, застосовуючи властивості числа сполук (Таблиця1, Таблиця2).

А всі інші ми будемо заповнювати, обчислюючи числа сполук з використанням мови програмування.

Таблиця 1

| | | | | | | | | | | | | | | | | |
|-----|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|--|---------|--|--|--|--|
| n=0 | | | | | | | | C_0^0 | | | | | | | | |
| n=1 | | | | | | | | C_1^0 | C_1^1 | | | | | | | |
| n=2 | | | | | | | C_2^0 | C_2^1 | C_2^2 | | | | | | | |
| n=3 | | | | | C_3^0 | | C_3^1 | C_3^2 | C_3^3 | | | | | | | |
| n=4 | | | | C_4^0 | | C_4^1 | C_4^2 | C_4^3 | C_4^4 | | | | | | | |
| n=5 | | | C_5^0 | | C_5^1 | | C_5^2 | C_5^3 | C_5^4 | C_5^5 | | | | | | |
| n=6 | | C_6^0 | | C_6^1 | | C_6^2 | C_6^3 | C_6^4 | C_6^5 | C_6^6 | | | | | | |
| n=7 | C_7^0 | | C_7^1 | | C_7^2 | | C_7^3 | C_7^4 | C_7^5 | C_7^6 | | C_7^7 | | | | |

Таблиця 2

| | | | | | | | | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|--|
| n=0 | | | | | | | | 1 | | | | | | | | |
| n=1 | | | | | | | 1 | 1 | | | | | | | | |
| n=2 | | | | | 1 | | 2 | 1 | | | | | | | | |
| n=3 | | | | 1 | | 3 | | 3 | 1 | | | | | | | |
| n=4 | | | 1 | | 4 | | | 4 | 1 | | | | | | | |
| n=5 | | | 1 | | 5 | | | | 5 | 1 | | | | | | |
| n=6 | | 1 | | 6 | | | | | | 6 | 1 | | | | | |
| n=7 | 1 | | 7 | | | | | | | | 7 | 1 | | | | |

4. Допоміжні алгоритми та підпрограми.

Перед тим як скласти програму, поговоримо про підпрограму, яку ми будемо використовувати.

Давайте згадаємо визначення поняття «Підпрограма». Відповідь: Підпрограма – це допоміжний алгоритм, який записаний на мові програмування.

Що таке допоміжний алгоритм? Відповідь: Можна використовувати готовий алгоритм в іншому алгоритмі. Для цього необхідно зробити звернення до допоміжного алгоритму. В основному алгоритмі відбувається передача керування з поверненням. Цей процес бачимо на такому прикладі, як «переклад іноземного тексту»: зустрічаючи у тексті незнайоме слово, людина припиняє роботу в тексті і переходить на роботу зі словником, шукає переклад потрібного слова, знайшовши, вона повертається в те місце тексту, де перервалася, і продовжує роботу з текстом, зустрівши незнайоме слово, він знову звертається до словника і т. д.

Повернемося до обчислення числа сполучень. Обчислення числа сполучень зводиться до розв’язання задачі за допомогою комп’ютера.

Обчислити значення $CNK = \frac{n!}{k!(n-k)!}$, де, N - кількість об'єктів, з яких робиться вибір, K - кількість елементів у комбінації.

Розв'язання: 1 етап. Постановка задачі.

Дано: N, K - значення аргументу

Знайти: CNK - ?

2 етап. Математична формалізація. Відразу обчислити ми не можна, так як нам дано значення N і K, а у формулі використовується N!, R!, (N - K)!, тобто потрібно ввести проміжні змінні для цих значень. Позначимо, A=N!, Тоді A = 1 * 2 * 3 * * N,

B = K!, Тоді B = 1 * 2 * 3 * * K,

C = (N - K)!, Тоді C = 1 * 2 * 3 * (N - K).

Тепер наша формула прийме вигляд: $CNK = A / (B * C)$.

При складанні алгоритму ми будемо використовувати допоміжний алгоритм. З математичної моделі видно, що факторіал обчислюється тричі, тому очевидно, що допоміжний алгоритм - це обчислення факторіала. Вам відомий алгоритм обчислення M! Будемо використовувати його в якості допоміжного алгоритму.

3 етап. Побудова блок- схеми.

Побудуйте блок- схему обчислення факторіала.

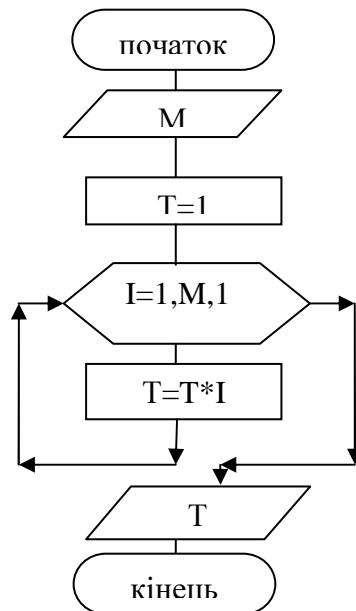


Рис. 1. Блок-схема допоміжного алгоритму

Перший раз необхідно обчислити N!, А в допоміжному алгоритмі M!, Необхідно виконати M = N, це привласнення потрібно зробити в основному алгоритмі, перед викликом допоміжного алгоритму.

Результат роботи допоміжного алгоритму у змінній T, а в математичній моделі значення N! повинно бути змінною A, тобто в основному алгоритмі треба присвоїти $A = T$. Вихідні дані для допоміжного алгоритму задаються в основному алгоритмі і результат роботи допоміжного алгоритму використовуються в основному алгоритмі [1].

5. Складання програми. Які види підпрограм в мові програмування Pascal вам відомі? Відповідь: процедури та функції.

Використовуючи процедуру або функцію складемо алгоритм програми на мові програмування Pascal.

4 етап: Складемо програму [2].

Program Tablicja;

Var N, K: integer;

A, B, C, CNK: integer;

Procedure Spoluka (m: integer; var t: integer);

Var i:integer;

Begin

t:=1;

for i:=1 to m do { алгоритм роботи підпрограми }

t:=t+1;

End;

Begin

Writeln ('Введіть N, K');

Readln (N,K);

Spoluka (N, A);

Spoluka (K, B);

Spoluka (N - K, C);

CNK:= A / (B * C).

Writeln('CNK=', CNK);

End.

6. Заповнення таблиці. А тепер виконайте програму, обчисліть значення і заповніть таблицю до 7 рядка включно. (1 варіант заповнює 4, 7; 2 варіант - 5,6). У результаті виходить:

| | | | | | | | | | | | |
|---------|--|--|--|---|---|----|----|----|----|---|---|
| 0 рядок | | | | 1 | | | | | | | |
| 1 | | | | 1 | 1 | | | | | | |
| 2 | | | | 1 | 2 | 1 | | | | | |
| 3 | | | | 1 | 3 | 3 | 1 | | | | |
| 4 | | | | 1 | 4 | 6 | 4 | 1 | | | |
| 5 | | | | 1 | 5 | 10 | 10 | 5 | 1 | | |
| 6 | | | | 1 | 6 | 15 | 20 | 15 | 6 | 1 | |
| 7 | | | | 1 | 7 | 21 | 35 | 35 | 21 | 7 | 1 |

Дану таблицю можна продовжувати і далі. Отримана таблиця називається трикутником Паскаля по імені французького вченого Блеза Паскаля (1626-1662), у працях якого цей трикутник зустрічається. Але історично це назва не зовсім точна, справа в тому, що в XIII столітті Персицький вчений Омар Хайям вже знав про існування такої таблиці. Більш правильно його називати «арифметичний трикутник».

7. Виявлення ще одної властивості сполук.

Якщо уважно подивитися, то можна помітити певну закономірність між числами таблиці. Ця закономірність виражається у вигляді ще однієї властивості сполук:

$$C_n^k + C_n^{k+1} = C_{n+1}^{k+1}$$

Для обґрунтування даної рівності знайдемо суму $C_n^k + C_n^{k+1}$, враховуючи, що

$$(k+1)! = 1 \cdot 2 \cdot \dots \cdot k \cdot (k+1) = k!(k+1),$$

$$(n-k)! = 1 \cdot 2 \cdot \dots \cdot (n-k-1) \cdot (n-k) = (n-k-1)!(n-k).$$

$$\begin{aligned} C_n^k + C_n^{k+1} &= \frac{n!}{k!(n-k)!} + \frac{n!}{(k+1)!(n-k-1)!} = \frac{n!}{k!(n-k-1)!(n-k)} + \frac{n!}{k!(k+1)(n-k-1)!} = T \\ &= \frac{n!(k+1+n-k)}{(k+1)!(n-k)!} = \frac{n!(n+1)}{(k+1)!(n-k)!} = \frac{(n+1)!}{(k+1)!(n-k)!}. \end{aligned}$$

$$A \quad C_{n+1}^{k+1} = \frac{(n+1)!}{(k+1)!((n+1)-(k+1))!} = \frac{(n+1)!}{(k+1)!(n-k)!}$$

$$\text{обто } C_n^k + C_n^{k+1} = C_{n+1}^{k+1}.$$

Таким чином, щоб отримати, наприклад, четвертий рядок таблиці Паскаля треба на першому місці рядка записати одиницю. На другому місці запишемо число, рівне сумі двох чисел третього рядка, що стоять над ним лівіше і правіше (оскільки за властивістю $C_4^1 = C_3^0 + C_3^1 = 1 + 3 = 4$). На третьому місці запишемо число, яке рівне сумі двох наступних чисел третього рядка, які стоять над ним лівіше і правіше ($C_4^2 = C_3^1 + C_3^2 = 3 + 3 = 6$) і т. д. (а на останньому місці знову запишемо одиницю) [3].

8. Підсумки.

Не випадково арифметичному трикутнику приділяється велика увага, він цього заслуговує. У цьому ви переконаєтеся на наступних заняттях комбінаторики. Ще не раз ми будемо звертатися до нього, до закономірностей і числах, з яких він складений.

Сьогодні ви ще раз побачили, що за допомогою комп'ютера можна розв'язувати комбінаторні задачі різного роду і різного рівня складності. При цьому головне правильно використовувати ресурси мови програмування. Допоміжний алгоритм - це зручний засіб, який полегшує розв'язання важких задач, і дійсно складний алгоритм, як би розбивається на простіші-допоміжні

алгоритми, т. б. розв'язання ведеться від простого до складного, а в цьому і полягає мистецтво складання алгоритмів. При цьому алгоритм стає лаконічним, компактним, наочним. Ви будете продовжувати вчитися розв'язувати комбінаторні задачі із застосуванням мови програмування.

Домашнє завдання: Скласти алгоритм розв'язання наступної комбінаторної задачі: скільки існує різних способів розміняти одну гривну різними монетами.

Висновки. Таким чином був представлений можливий варіант конспекту інтегрованого заняття з математики з елементами основ програмування. Даний спосіб подання матеріалу допоможе студентам не тільки краще засвоїти практичну частину подання матеріалу, на етапі розв'язання задач, а й підтримає зв'язок з майбутньою спеціальністю вже під час вивчення дисциплін загальноосвітнього циклу.

Все вище сказане зумовлює необхідність подальшого вивчення способів реалізації міждисциплінарних зв'язків між дисциплінами загальноосвітнього, природничо-наукового циклів та спеціальних дисциплін, реалізуючи принципи інтеграції навчального матеріалу.

Література

- 1.Разработка 2 –х уроков по теме: «Построение треугольника Паскаля с применением средств языка программирования QBASIC». [Электронный ресурс] – Режим доступа: <http://uch.znate.ru/docs/1417/index-10230.html>
- 2.Практический курс программирования на ПАСКАЛЕ. [Текст] / М.А. Черкасов - Учебное пособие. Издательство: Москва Год: 2005. — 180с.
- 3.Нелин, Е.П., Долгова О.Е. Алгебра и начала анализа: Двухуровневый учеб. для 11 кл. общеобразоват. учеб. заведений [Текст] / Пер. с укр. Е. П. Нелина.— Х.: Мир детства, 2006.— 416 с.