

Задания на множества, словари и комбинаторику в PascalABC.NET

Михалкович С.С.

*Южный федеральный университет,
Институт математики, механики и компьютерных наук*
miks@sfedu.ru

Доклад на пятой Всероссийской научно-методической конференции
«Использование системы программирования PascalABC.NET
в обучении программированию»
(Ростов-на-Дону, 28-29 марта 2025 г.)

Важнейшие коллекции в PascalABC.NET

В PascalABC.NET, как и в других современных языках программирования, есть несколько основных типов коллекций: **множества**, **словари** и **списки**. Каждая из них имеет свои особенности и применяется для решения определённых задач.

- **Списки** — это коллекции, где каждый элемент имеет свой индекс. Списки похожи на массивы, однако могут эффективно расширяться, а также позволяют вставлять и удалять элементы.
- **Множества** — это коллекции, которые хранят только уникальные элементы. Основные операции: добавление, удаление и проверка на принадлежность. Доступны математические операции: объединение, пересечение и разность.
- **Словари** — это коллекции, которые хранят данные в виде пар "ключ-значение". Они позволяют быстро находить значение по ключу и добавлять новые элементы с использованием синтаксиса, похожего на массивы:
`d[ключ] := значение.`

Проблемы при изучении коллекций

Основная сложность при изучении **множеств, словарей** и **списков** заключается в подборе подходящих задач, которые:

1. Соответствуют уровню обучающихся.
2. Раскрывают суть каждой коллекции.
3. Закрепляют навыки работы с различными типами данных.
4. Включают короткие прикладные задачи, чтобы показать практическую пользу.

Кроме того, задачи должны:

- Сочетать использование нескольких коллекций вместе.
- Для **множеств** — активно включать задачи на **комбинаторику**, которая поддерживается в PascalABC.NET встроенными методами (например, генерация перестановок, сочетаний и размещений).

Множества – задачи

- Создание множеств. Объединение, пересечение, разность. Равенство множеств.
- Объединение и пересечение 3 и более множеств
- Множества строк, множества полностью отличаются (пересечение пусто)
- Проверка на принадлежность множеству, not in. Массив множеств
- Добавление элемента во множество – удаление из множества
- Преобразование массива во множество. Определение уникальных элементов и их количества
- Количество различных слов в строке
- Буквы, присутствующие в каждом из трех слов
- Операция вложения, строгого вложения. Множество цифр в строке
- Города, которые посещал только один ученик

Объединение, пересечение, разность, равенство

Создание множеств. Объединение,
пересечение, разность.

Равенство множеств

```
begin
  var m := SetOf(5,2,8,3,9);
  var p := SetOf(4,8,9,7,5);
  Println(m * p);
  Println(m + p);
  Println(m - p);
  Println(p - m);

  var m := SetOf(15,28,53,44,29);
  var p := SetOf(53,29,15,28,44);
  Print(m = p);
end.
```

```
{5,8,9}
{5,2,8,3,9,4,7}
{2,3}
{4,7}
True
```

Непересекающиеся множества

Множества не пересекаются если
их пересечение пусто
(пересечение равно пустому
множеству)

```
begin
  var s1 :=
    SetOf('красный', 'зеленый', 'синий', 'желтый');
  var s2 := SetOf('желтый', 'оранжевый',
    'фиолетовый');
  var s3 := SetOf('оранжевый', 'черный',
    'голубой', 'фиолетовый');

  if s1 * s2 = [] then
    Println(s1, '-', s2);
  if s1 * s3 = [] then
    Println(s1, '-', s3);
  if s2 * s3 = [] then
    Println(s2, '-', s3);
end.
```

{красный, зеленый, синий, желтый} -
{оранжевый, черный, голубой, фиолетовый}

Проверка на принадлежность к множеству

В скольких множествах есть
зеленый, но нет желтого?

```
begin
  var s1 :=
    SetOf( 'красный', 'зеленый', 'синий', 'желтый' );
  var s2 :=
    SetOf( 'фиолетовый', 'синий', 'красный', 'оранжевый' );
  var s3 := SetOf( 'синий', 'оранжевый', 'зеленый' );
  var s4 := SetOf( 'желтый', 'зеленый', 'голубой' );
  var a := Arr(s1,s2,s3,s4);
  var count := 0;
  foreach var s in a do
    if ( 'зеленый' in s ) and ( 'желтый' not in s ) then
      count += 1;
  Print(count);
end.
```

1

Добавление и удаление элементов

Добавить три случайных и удалить
все нечетные

```
begin
  var s := SetOf(1, 3, 5, 6, 9);
  loop 3 do
    s.Add(Random(10,20));
    Println(s);
    foreach var x in s do
      if x.IsOdd then
        s.Remove(x);
    Println(s);
  end.
```

{1,3,5,6,9,20,10,12}

{6,20,10,12}

Преобразование массива во множество

Сколько в массиве различных элементов?

Превратим массив во множество – дубли исчезнут

```
begin
  var a := Arr(1, 5, 5, 4, 3, 2, 6, 5, 7,
3, 8, 3, 6);
  var s := a.ToSet;
  Print(s.Count);
end.
```

8

Количество различных слов в строке

Сколько в данной строке
содержится различных слов

```
begin  
  var s := 'she sells sea shells sea shells  
she sells sea shells';  
  s.ToWords.ToSet.Count.Print  
end.
```

4

Буквы, присутствующие в каждом слове

Какие буквы присутствуют во всех
трех словах

```
begin
  var s1 := 'стрекоза';
  var s2 := 'барокамера';
  var s3 := 'программирование';
  Print(s1.ToSet * s2.ToSet * s3.ToSet)
end.
```

{p,e,o,a}

Множество цифр в числе

Проверьте, что все цифры первого числа есть во втором числе

```
begin
  var x := 1668922;
  var y := 90192697672838;
  var sx := x.ToString.ToSet;
  var sy := y.ToString.ToSet;
  Print(sx <= sy);
end.
```

True

Города, которые посещал только один ученик

Вывести города, которые посещал
только один ученик

```
begin
  var Маша := 'Москва Петербург Пятигорск
Москва Волгоград';
  var Петя := 'Волгоград Москва Смоленск
Москва Волгоград';
  var s1 := Маша.ToWords.ToSet;
  var s2 := Петя.ToWords.ToSet;
  Print((s1 - s2) + (s2 - s1))
end.
```

{Петербург,Пятигорск,Смоленск}

Комбинаторика – задачи

Основное в этих задачах. В подавляющем большинстве материалов по комбинаторике считается количество – мы же благодаря этим задачам можем увидеть **сами комбинаторные объекты**

- **Декартово произведение множеств** учеников (мальчиков и девочек).
Вложенные циклы и метод Cartesian
- **Декартова степень множеств** цифр. Вложенные циклы или метод CartesianPower
- Комбинации букв слова, удовлетворяющие некоторым ограничениям
- **Перестановки** чисел, их количество, большие числа количества перестановок (факториал)
- Перестановки с ограничениями, их количество
- Анаграммы, ручной выбор
- **Частичные перестановки** (размещения)
- **Сочетания** и их отличие от частичных перестановок
- Прикладные задачи - букеты тюльпанов разных цветов
- Сочетания с ограничениями
- **Множество всех подмножеств** – кто пошел на субботник – все варианты

Декартово произведение множеств

Вывести всевозможные пары
мальчик девочка

```
begin
  var boys := Arr('Петя', 'Коля');
  var girls := Arr('Марина', 'Таня', 'Ира');
  foreach var boy in boys do
    foreach var girl in girls do
      Println(boy, girl);

  boys.Cartesian(girls).Print
end.
```

```
Петя Марина
Петя Таня
Петя Ира
Коля Марина
Коля Таня
Коля Ира
```

Декартова степень множеств

Составьте все трехзначные числа из цифр 5,4,2,7

```
begin
  var a := Arr(5,4,2,7);
  foreach var dd in a.CartesianPower(3) do
    Write(dd[0],dd[1],dd[2], ' ')
  end.
```

```
555 554 552 557 545 544 542 547 525 524 522 527
575 574 572 577 455 454 452 457 445 444 442 447
425 424 422 427 475 474 472 477 255 254 252 257
245 244 242 247 225 224 222 227 275 274 272 277
755 754 752 757 745 744 742 747 725 724 722 727
775 774 772 777
```

Комбинации букв с ограничениями

Составьте все слова длины 4 из букв слова 'рак', содержащие не более 2 букв 'а'

```
begin
  foreach var word in
    'рак'.CartesianPower(4) do
    if word.CountOf('а') <= 2 then
      Print(word)
end.
```

```
pppp ppra prrk rpar rраа ррак рркp ррка рркк
rapp рара парк раар раак ракр рака раkk ркрp
ркра рркк ркар ркаа ркак рккр ркка рккк appp
appa appk араp араk аркр арка аркк аapp аарк
аакр аакк акpp акра акрк акар акак аккр akka
аккк кppp кppa кppk кpар кpаа кpак кркp крка
кркк кarp кара карк каар каак какр кака какк
ккpp ккра кркк ккар ккаа ккак кккр ккка кккк
```

Перестановки

Выведите все перестановки чисел
1,2,3,4

```
begin  
  var a := Arr(1,2,3,4);  
  a.Permutations.Print  
end.
```

```
[1,2,3,4] [1,2,4,3] [1,3,2,4] [1,3,4,2]  
[1,4,2,3] [1,4,3,2] [2,1,3,4] [2,1,4,3]  
[2,3,1,4] [2,3,4,1] [2,4,1,3] [2,4,3,1]  
[3,1,2,4] [3,1,4,2] [3,2,1,4] [3,2,4,1]  
[3,4,1,2] [3,4,2,1] [4,1,2,3] [4,1,3,2]  
[4,2,1,3] [4,2,3,1] [4,3,1,2] [4,3,2,1]
```

Перестановки с ограничениями

Сколько существует перестановок из чисел 1..8, у которых число 8 стоит на третьем месте?

```
begin
  var a := Arr(1..8);
  a.Permutations.Count(p -> p[2]=8).Print
end.
```

5040

Анаграммы

Выведите все слова, полученные перестановкой букв в слове 'сорт'

Найдите среди них правильные русские слова. Они называются **анаграммами**

```
begin  
  'сорт'.Permutations.Print  
end.
```

```
сорт сотр срот срто стор стро осрт остр орст  
ортс отср отрс рсот рсто рост ротс ртсо ртос  
тсор тсро тоср торс трсо трос
```

Частичные перестановки

Выведите все трехбуквенные слова, которые можно построить, используя буквы слова 'строка'.

Каждую букву можно использовать только один раз

```
begin  
  'строка'.Permutations(3).Print  
end.
```

стр	срт	тср	трс	рст	ркс	сто	сот	тсо	тос	ост	отс
стк	скт	тск	ткс	кст	ктс	ста	сат	тса	тас	аст	атс
сро	сор	рсо	рос	оср	орс	срк	скр	рск	ркс	ксп	крс
сра	сар	рса	рас	аср	арс	сок	ско	оск	окс	ксо	кос
соа	сао	оа	оас	асо	аос	ска	сак	кса	кас	аск	акс
тро	тор	рто	рот	отр	орт	трк	ткр	ррк	ркт	ктр	крт
тра	тар	рта	рат	атр	арт	ток	тко	отк	окт	кто	кот
тоа	тао	ота	оат	ато	аот	тка	так	кта	кат	атк	акт
рок	рко	орк	окр	кро	кор	роа	рао	ора	оар	аро	аор
рка	рак	кра	кар	арк	акр	ока	оак	коа	као	аок	ако

Сочетания. Букеты тюльпанов

В магазине - тюльпаны пяти цветов

- красного, белого, розового,
желтого и малинового

Выведите все варианты букетов
из тюльпанов разных цветов

```
begin
  var tulips := Arr('красный', 'белый',
    'розовый', 'желтый', 'малиновый');
  tulips.Combinations(3).Print
end.
```

```
[красный, белый, розовый]
[красный, белый, желтый]
[красный, белый, малиновый]
[красный, розовый, желтый]
[красный, розовый, малиновый]
[красный, желтый, малиновый]
[белый, розовый, желтый]
[белый, розовый, малиновый]
[белый, желтый, малиновый]
[розовый, желтый, малиновый]
```

Сочетания с фильтрацией

Выведите все комбинации из
четырех разных цифр, сумма
которых равна 20
Порядок цифр неважен!

```
begin
  var a := Arr(0..9);
  a.Combinations(4)
    .Where(a -> a.Sum = 20).Print
end.
```

```
[0,3,8,9] [0,4,7,9] [0,5,6,9] [0,5,7,8]
[1,2,8,9] [1,3,7,9] [1,4,6,9] [1,4,7,8]
[1,5,6,8] [2,3,6,9] [2,3,7,8] [2,4,5,9]
[2,4,6,8] [2,5,6,7] [3,4,5,8] [3,4,6,7]
```

Множество всех подмножеств

Дано множество учеников.

Вывести все подмножества
этого множества.

```
begin
  var pupils :=
    Arr('Вася', 'Оля', 'Ира', 'Петя');
  for var i:=0 to pupils.Count do
    pupils.Combinations(i).Println
end.
```

```
[]
[Вася] [Оля] [Ира] [Петя]
[Вася,Оля] [Вася,Ира] [Вася,Петя] [Оля,Ира]
[Оля,Петя] [Ира,Петя]
[Вася,Оля,Ира] [Вася,Оля,Петя]
[Вася,Ира,Петя] [Оля,Ира,Петя]
[Вася,Оля,Ира,Петя]
```

Словари – задачи

- Словарь переводов. Поиск слова в словаре
- Частота слов в строке
- Словарь оценок учеников. Упорядочение его по оценкам и по фамилиям
- Обратный словарь
- Объединение словарей – телефонных книг – в единый словарь
- Словарь учеников и их оценок. Интеграция в словарь средних оценок
- Фильтрация значений в словаре
- Преобразование словаря в словарь с преобразованными значениями
- Преобразование массива пар в словарь – создание объектов по категориям
- Словарь преобразований слов и замена этих слов в тексте
- Словарь игроков и их очков – очки меняются в процессе игры, в конце выводится Топ-3 игроков

Словарь переводов

Реализовать простой переводчик слов с английского на русский.

```
begin
  var d := new Dictionary<string,string>;
  d['hello'] := 'привет';
  d['world'] := 'мир';
  d['sun'] := 'солнце';
  var word := ReadString;
  if word in d then
    Print('${word} переводится как {d[word]}')
  else Print('${Перевод для {word} не найден');
end.
```

Частота слов в строке

Дана строка слов. По ней
составьте частотный словарь

```
begin
  var s := ReadString;
  var words := s.ToWords;
  var d := new Dictionary<string,integer>;
  foreach var word in words do
    if word in d then
      d[word] += 1
    else d[word] := 1;
  foreach var pair in d do
    Println(pair.Key,pair.Value)
end.
```

Обратный словарь

Дан словарь. По нему составить обратный словарь и вывести

```
begin
  var original
    := new Dictionary<string,integer>;
  original['один'] := 1;
  original['два'] := 2;
  original['четыре'] := 4;

  var inversed
    := new Dictionary<integer,string>;

  foreach var pair in original do
    inversed[pair.Value] := pair.Key;

  inversed.Print
end.
```

(1,один) (2,два) (4,четыре)

Объединение словарей

Даны два словаря с телефонными книгами. Объединить их в один. При наличии записей в обоих словарях учитывать последний номер телефона.

(Мария, 987-6543) (Иван, 765-4321)
(Алексей, 555-1234)

```
begin
  var phoneBook1
    := new Dictionary<string, string>;
  phoneBook1['Мария'] := '987-6543';
  phoneBook1['Иван'] := '123-4567';
  var phoneBook2
    := new Dictionary<string, string>;
  phoneBook2['Иван'] := '765-4321';
  phoneBook2['Алексей'] := '555-1234';

  var united := new Dictionary<string, string>;
  foreach var pair in phoneBook1 do
    united[pair.Key] := pair.Value;
  foreach var pair in phoneBook2 do
    united[pair.Key] := pair.Value;

  united.Print
end.
```

Словарь средних оценок

Задание. Дан словарь учеников и их оценок, заданных в виде массива.

По нему создать словарь учеников и для каждого вычислить средний балл.

```
begin
  var studentGrades:
    Dictionary<string, array of integer>
    := Dict(
      'Иван' to [5, 4, 4, 3],
      'Мария' to [4, 4, 5, 5],
      'Алексей' to [3, 3, 4, 4],
      'Ольга' to [5, 5, 5, 4]
    );
  var avgGrades := new Dictionary<string, real>;
  foreach var pair in studentGrades do
    avgGrades[pair.Key] := pair.Value.Average;

  averageGrades.Println
end.
```

(Иван,4) (Мария,4.5) (Алексей,3.5) (Ольга,4.75)

Фильтрация значений в словаре

Дан словарь учеников и их средние баллы.

Создайте словарь учеников, у которых средний балл выше заданного порога.

```
begin
  var scores := Dict('Иван' to 85,
    'Мария' to 7,
    'Алексей' to 34,
    'Ольга' to 57);

  var x := ReadInteger;
  var newscores
    := new Dictionary<string, integer>;
  foreach var pair in scores do
    if pair.Value > x then
      newscores[pair.Key] := pair.Value;
  newscores.Print;
end.
```

Преобразование словаря

Дан словарь городов и средних температур в градусах Цельсия. Создайте такой же словарь, в котором градусы Цельсия будут заменены на градусы Кельвина

```
begin
  var temperaturesC := Dict('Москва' to 15,
    'Лондон' to 11, 'Нью-Йорк' to 23);

  var q := temperaturesC.Select(p ->
    (p.Key to (p.Value * 9 / 5 + 32)));

  var d := Dict(q);

  d.Println;
end.
```

Москва,59) (Лондон,51.8) (Нью-Йорк,73.4)

Создание объектов по категориям

Дан массив пар (название
продукта, тип продукта).

Сгруппируйте массив пар, сделав
тип продукта ключем словаря, а
значением – список этих
продуктов

фрукт	[яблоко, банан, манго]
овощ	[морковь, огурец]

```
begin
  var items := [('яблоко', 'фрукт'),
    ('морковь', 'овощ'),
    ('банан', 'фрукт'),
    ('огурец', 'овощ'),
    ('манго', 'фрукт')];

  var d := new Dictionary<string, List<string>>;
  foreach var (name, category) in items do
    begin
      if category not in d then
        d[category] := new List<string>;
        d[category].Add(name)
      end;
    foreach var pair in d do
      Println(pair.Key, pair.Value);
    end.
end.
```

Словарь преобразований слов

Заменить в данном тексте все
слова из словаря на
соответствующие им слова

```
begin
  var emoji := Dict(
    'улыбка' to $'😊',
    'солнце' to $'☀️',
    'дождь' to $'🌧️',
    'кот' to $'🐱',
    'сердце' to $'❤️'
  );

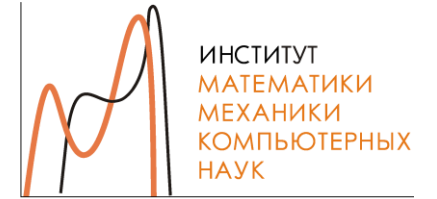
  var message := 'Я и кот , солнце в душе и прекрасная
улыбка в моем сердце';
  foreach var word in message.ToWords do
    if word in emoji then
      Print(emoji[word])
    else Print(word)
  end.
```

Я и 🐱 , ☀️ в душе и прекрасная 😊 в моем ❤️

Выводы

Выводы

- PascalABC.NET позволяет легко использовать такие **классы коллекций** как списки, массивы, множества и словари
- При этом становится возможно легко решать ряд **прикладных задач**, которые ранее не решались на штатных занятиях со школьниками ввиду высокой сложности
- Встроенные методы работы с **комбинаторными объектами** позволяют легко визуализировать перестановки,, перестановки с повторениями, сочетания, а также декартово произведение и декартову степень множеств, а также использовать эти комбинаторные объекты в прикладных задачах



Михалкович С.С.

*Южный федеральный университет,
Институт математики, механики и компьютерных наук*
miks@sfedu.ru

