

# Новые исполнители и новые элементы программирования при обучении школьников

Михалкович С.С.

*Южный федеральный университет,  
институт математики,  
механики и компьютерных наук*

[miks@math.sfedu.ru](mailto:miks@math.sfedu.ru)

Доклад на

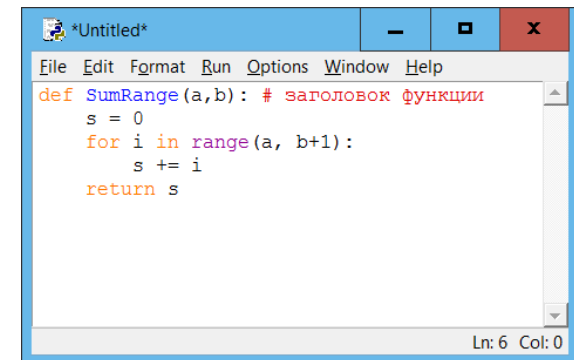
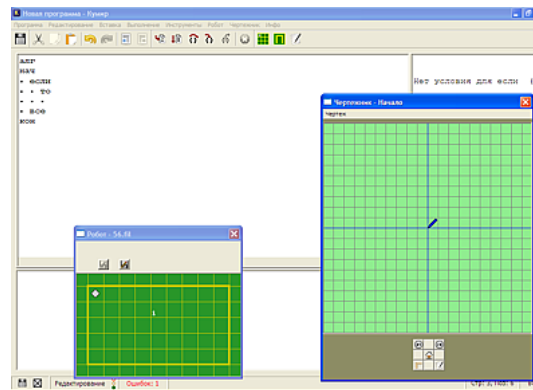
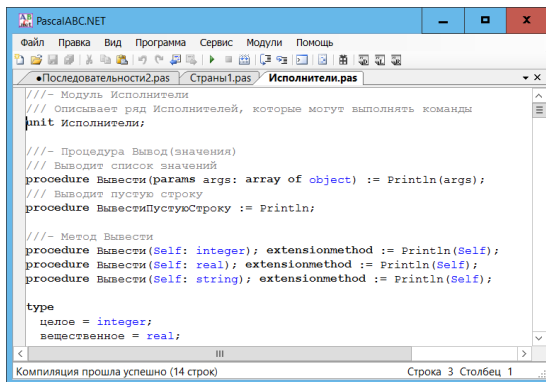
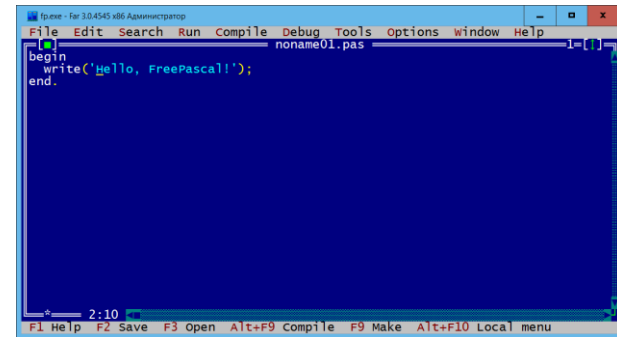
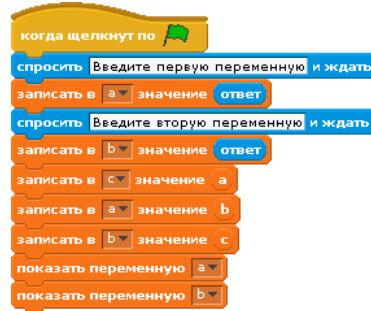
V Международном научно-методическом симпозиуме  
«Электронные ресурсы в непрерывном образовании»

(ЭРНО-2016)

Анапа, 26-29 сентября 2016 г.

# Языки для обучения программированию школьников

- Scratch
- КуМир
- Python
- Free Pascal
- PascalABC.NET



# О PascalABC.NET

- Начало проекта PascalABC.NET – **2006 г.**
- Среда и язык программирования, пришедшие на замену «стандартному Паскалю»
- Высокая популярность, использование при обучении в школах и ВУЗах
- Наличие веб-среды разработки (Паскаль в браузере)
- Встроенный электронный задачник Programming Taskbook

# Рост популярности

Этап всероссийской олимпиады по информатике в Москве

	2015-16						2014-15					
	Школьный		Муниципальный		Региональный		Школьный		Муниципальный		Региональный	
Всего участников (>0 баллов)	7371	100,00%	2467	100,00%	435	100,00%	5740	100,00%	1932	100,00%	478	100,00%
<b>PascalABC.NET</b>	3782	51,31%	1150	<b>46,62%</b>	64	14,71%	2723	47,44%	552	<b>28,57%</b>	59	12,34%
FPC	1040	14,11%	166	<b>6,73%</b>	29	6,67%	881	15,35%	433	<b>22,41%</b>	58	12,13%
Дельфи	27	0,37%	2	0,08%	4	0,92%	43	0,75%	30	1,55%	9	1,88%
Все паскали	4812	65,28%	1306	<b>52,94%</b>	85	19,54%	3608	62,86%	944	48,86%	102	21,34%
g++	747	10,13%	432	17,51%	211	48,51%	557	9,70%	368	19,05%	222	46,44%
gcc	242	3,28%	100	4,05%	13	2,99%	225	3,92%	87	4,50%	22	4,60%
clang++					9	2,07%	34	0,59%	20	1,04%	9	1,88%
clang							8	0,14%	4	0,21%	4	0,84%
Все C/C++	971	13,17%	530	21,48%	231	53,10%	811	14,13%	462	23,91%	244	51,05%
Python-3	859	11,65%	407	16,50%	151	34,71%	580	10,10%	379	19,62%	174	36,40%
Python-2	76	1,03%	24	0,97%	3	0,69%	38	0,66%	22	1,14%	8	1,67%
Все питоны	925	12,55%	429	17,39%	153	35,17%	611	10,64%	395	20,45%	179	37,45%
Кумир-1	219	2,97%	43	1,74%	0	0,00%	308	5,37%	31	1,60%	0	0,00%
Кумир-2	162	2,20%	26	1,05%	0	0,00%	77	1,34%	18	0,93%	0	0,00%
Все кумиры	373	5,06%	68	2,76%	0	0,00%	378	6,59%	44	2,28%	0	0,00%
Qbasic (fbc)	116	1,57%	13	0,53%	1	0,23%	240	4,18%	20	1,04%	0	0,00%
Visual Basic	36	0,49%	14	0,57%	0	0,00%	34	0,59%	10	0,52%	2	0,42%
FBC-32	16	0,22%	6	0,24%	0	0,00%	3	0,05%	0	0,00%	0	0,00%
Все бейсики	168	2,28%	33	1,34%	1	0,23%	277	4,83%	30	1,55%	2	0,42%
C#	164	2,22%	88	3,57%	7	1,61%	122	2,13%	62	3,21%	15	3,14%
Java	120	1,63%	65	2,63%	9	2,07%	58	1,01%	46	2,38%	13	2,72%
php	28	0,38%	6	0,24%	0	0,00%	18	0,31%	7	0,36%	1	0,21%
perl	2	0,03%	2	0,08%	0	0,00%	7	0,12%	3	0,16%	1	0,21%
ruby	3	0,04%	2	0,08%	0	0,00%	0	0,00%	1	0,05%	1	0,21%

# Рост популярности

Этап всероссийской олимпиады по информатике в Москве

	2015-16						2014-15					
	Школьный		Муниципальный		Региональный		Школьный		Муниципальный		Региональный	
Всего участников (>0 баллов)	7371	100,00%	2467	100,00%	435	100,00%	5740	100,00%	1932	100,00%	478	100,00%
<b>PascalABC.NET</b>	3782	51,31%	1150	<b>46,62%</b>	64	14,71%	2723	47,44%	552	<b>28,57%</b>	59	12,34%
FPC	1040	14,11%	166	<b>6,73%</b>	29	6,67%	881	15,35%	433	<b>22,41%</b>	58	12,13%
Дельфи	27	0,37%	2	0,08%	4	0,92%	43	0,75%	30	1,55%	9	1,88%
Все паскали	4812	65,28%	1306	<b>52,94%</b>	85	19,54%	3608	62,86%	944	48,86%	102	21,34%
g++	747	10,13%	432	17,51%	211	48,51%	557	9,70%	368	19,05%	222	46,44%
gcc	242	3,28%	100	4,05%	13	2,99%	225	3,92%	87	4,50%	22	4,60%
clang++					9	2,07%	34	0,59%	20	1,04%	9	1,88%
clang							8	0,14%	4	0,21%	4	0,84%
Все C/C++	971	13,17%	530	21,48%	231	53,10%	811	14,13%	462	23,91%	244	51,05%
Python-3	859	11,65%				34,71%	580	10,10%	379	19,62%	174	36,40%
Python-2	76	1,04%				0,69%	38	0,66%	22	1,14%	8	1,67%
Все питоны							611	10,64%	395	20,45%	179	37,45%
Кумир-1							308	5,37%	31	1,60%	0	0,00%
Кумир-2							77	1,34%	18	0,93%	0	0,00%
Все кумиры							378	6,59%	44	2,28%	0	0,00%
Qbasic (fbc)							240	4,18%	20	1,04%	0	0,00%
Visual Basic							34	0,59%	10	0,52%	2	0,42%
FBC-32							3	0,05%	0	0,00%	0	0,00%
Все бейсики							277	4,83%	30	1,55%	2	0,42%
C#	164	2,22%	88	3,57%	7	1,61%	122	2,13%	62	3,21%	15	3,14%
Java	120	1,63%	65	2,63%	9	2,07%	58	1,01%	46	2,38%	13	2,72%
php	28	0,38%	6	0,24%	0	0,00%	18	0,31%	7	0,36%	1	0,21%
perl	2	0,03%	2	0,08%	0	0,00%	7	0,12%	3	0,16%	1	0,21%
ruby	3	0,04%	2	0,08%	0	0,00%	0	0,00%	1	0,05%	1	0,21%



# Количество скачиваний

2016 год -  
**3000** скачиваний в день

2013 год -  
**500** скачиваний в день

РАВС.NET по дням: РАВСFull.NET по дням: РАВСMini.NET по дням:

Дата	Количество	Дата	Количество	Дата	Количество
2016-03-05	281	2016-03-05	854	2016-03-05	98
2016-03-04	497	2016-03-04	1320	2016-03-04	163
2016-03-03	730	2016-03-03	2016	2016-03-03	277
2016-03-02	857	2016-03-02	2198	2016-03-02	314
2016-03-01	835	2016-03-01	2171	2016-03-01	327
2016-02-29	722	2016-02-29	2113	2016-02-29	277
2016-02-28	666	2016-02-28	1999	2016-02-28	273
2016-02-27	439	2016-02-27	1464	2016-02-27	180
2016-02-26	617	2016-02-26	1818	2016-02-26	243
2016-02-25	822	2016-02-25	2236	2016-02-25	299
2016-02-24	822	2016-02-24	2129	2016-02-24	286
2016-02-23	656	2016-02-23	1878	2016-02-23	249
2016-02-22	495	2016-02-22	1473	2016-02-22	167
2016-02-21	523	2016-02-21	1614	2016-02-21	188
2016-02-20	450	2016-02-20	1481	2016-02-20	171
2016-02-19	720	2016-02-19	2028	2016-02-19	245
2016-02-18	874	2016-02-18	2326	2016-02-18	298
2016-02-17	975	2016-02-17	2545	2016-02-17	344
2016-02-16	937	2016-02-16	2535	2016-02-16	366
2016-02-15	810	2016-02-15	2244	2016-02-15	253

Статистика скачиваний

Всего по ресурсам:

Тип ресурса	Количество
pabc	67026
pabcfull	144379
pabcmini	41416

Подробно: Дата:

РАВС.NET по дням: РАВСFull.NET по дням: РАВСMini.NET по дням:

Дата	Количество	Дата	Количество	Дата	Количество
2013-03-15	21	2013-03-15	52	2013-03-15	14
2013-03-14	108	2013-03-14	312	2013-03-14	108
2013-03-13	113	2013-03-13	315	2013-03-13	93
2013-03-12	111	2013-03-12	288	2013-03-12	88
2013-03-11	88	2013-03-11	331	2013-03-11	80
2013-03-10	83	2013-03-10	263	2013-03-10	77
2013-03-09	52	2013-03-09	155	2013-03-09	48
2013-03-08	59	2013-03-08	145	2013-03-08	37
2013-03-07	60	2013-03-07	163	2013-03-07	35
2013-03-06	105	2013-03-06	297	2013-03-06	91
2013-03-05	145	2013-03-05	341	2013-03-05	96
2013-03-04	114	2013-03-04	339	2013-03-04	94
2013-03-03	99	2013-03-03	324	2013-03-03	75
2013-03-02	96	2013-03-02	242	2013-03-02	62
2013-03-01	88	2013-03-01	301	2013-03-01	56
2013-02-28	112	2013-02-28	337	2013-02-28	85
2013-02-27	127	2013-02-27	382	2013-02-27	111
2013-02-26	118	2013-02-26	371	2013-02-26	108
2013-02-25	113	2013-02-25	337	2013-02-25	93
2013-02-24	101	2013-02-24	290	2013-02-24	75
2013-02-23	68	2013-02-23	172	2013-02-23	48
2013-02-22	95	2013-02-22	269	2013-02-22	55
2013-02-21	128	2013-02-21	325	2013-02-21	79

# Высокоуровневые исполнители в PascalABC.NET

- Исполнители, традиционно встречающиеся в программировании: Файл, Множество, Последовательность
- Исполнители, относящиеся к школьным предметам: Русский, География, Физика, Английский, Вычислитель
- Все вместе образуют **Мир исполнителей** – многообразный, с различными траекториями обучения

# Исполнители Вычислитель и Множество

## Вычислитель

```
begin  
  Вычислитель.РешитьКвадратноеУравнение (1, 3, 2);  
  Вывод.ПустаяСтрока;  
  Вычислитель.РешитьКвадратноеУравнение (1, 2, 2);  
  Вывод.ПустаяСтрока;  
  Вычислитель.ВывестиАрифметическуюПрогрессию (1, 2);  
  Вычислитель.ПлощадьРавностороннегоТреугольника (5).Вывести;  
end.
```

Квадратное уравнение:  $1*x*x+3*x+2=0$   
Решения:  $x1=-2$   $x2=-1$

Квадратное уравнение:  $1*x*x+2*x+2=0$   
Решений нет

Арифметическая прогрессия:  $a0=1$   $d=2$   
1 3 5 7 9 11 13 15 17 19

4.33012701892219

## Множество

```
uses Исполнители;  
  
begin  
  Множество.Добавить (5);  
  Множество.Добавить (3);  
  Множество.Добавить (5);  
  Множество.Добавить (4);  
  Множество.Вывести;  
  Множество.Удалить (3);  
  Множество.Вывести;  
  Множество.Содержит (3).Вывести;  
  Вывод.ПустаяСтрока;  
  Множество.ВывестиВсеМетоды;  
end.
```

3 4 5  
4 5  
False

Методы исполнителя Множество:

Добавить (элемент: целое)

Удалить (элемент: целое)

Вывести

Содержит (элемент: целое): логическое

ВывестиВсеМетоды



# Исполнитель Файл

## Переменные и автовывод типа

```
uses Исполнители;  
  
begin  
  Файл.ОткрытьНаЗапись ('текст.txt');  
  Файл.Записать ('Информатика');  
  Файл.Записать ('Математика');  
  Файл.Записать ('Русский');  
  Файл.Записать ('Физика');  
  Файл.Записать ('Русский');  
  Файл.Записать ('Информатика');  
  Файл.Закрыть;  
  Файл.ВывестиСодержимое ('текст.txt');  
end.
```

Содержимое файла текст.txt:

```
Информатика  
Математика  
Русский  
Физика  
Русский  
Информатика
```

# Исполнители – последовательности

## Переменные и автовывод типа

```
uses Исполнители;  
  
begin  
  var арифм := АрифметическаяПрогрессия (1, 2, 20);  
  арифм.Вывести;  
  арифм.Сумма.Вывести;  
  var геом := ГеометическаяПрогрессия (1, 2, 10);  
  геом.Вывести;  
  геом.Сумма.Вывести;  
  var случай := СлучайнаяПоследовательность (10, 1, 100);  
  случай.Вывести;  
  Вывести (случай.Минимум, случай.Максимум);  
end.
```

```
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39  
400  
1 2 4 8 16 32 64 128 256 512  
1023  
59 62 70 93 34 42 21 74 75 12  
12 93
```



# Методы последовательностей как новые примитивы

## Простейшие методы последовательностей

Последовательность .Сумма

Последовательность .Среднее

Последовательность .Минимум

Последовательность .Максимум

Последовательность .Вывести

Последовательность .ВывестиПострочно

Последовательность .Отсортировать

Последовательность .Взять (n)

Последовательность .Пропустить (n)



# Цикл foreach для последовательностей

## Методы последовательностей и цикл foreach

```
uses Исполнители;  
  
begin  
  var последовательность := СлучайнаяПоследовательность(10);  
  последовательность.Вывести;  
  
  последовательность.Сумма.Вывести;  
  
  var сумма := 0;  
  foreach var значение in последовательность do  
    сумма += значение;  
  сумма.Вывести  
end.
```

```
1 1 1 7 1 7 1 8 0 5  
32  
32
```

# Совместное использование исполнителей

## Исполнители Файл и МножествоСтрок

```
uses Исполнители;  
  
begin  
  var строки := Файл.ВсеСтроки('текст.txt');  
  строки.ВывестиПострочно;  
  
  foreach var строка in строки do  
    МножествоСтрок.Добавить(строка);  
  
  МножествоСтрок.Вывести;  
end.
```

Информатика  
Математика  
Русский  
Физика  
Русский  
Информатика

Информатика Математика Русский Физика

# Лямбды – современный примитив программирования

- Лямбды олицетворяют функциональный подход в программировании
- Позволяют избавиться от циклов в простых задачах
- В сочетании с запросами отделяют разные этапы решения задачи друг от друга

- Лямбды – функции:

`x -> x*x`

`страна -> страна.Название`

- Лямбды – действия:

`x -> Вывести(x)`

`() -> ОкноСписка.Отобразить(СтраныАзии)`

- `->` читается как «**преобразуется в**»

# Методы последовательностей с лямбдами

Последовательность .**Выбрать** (x -> x>2)  
Последовательность .**Преобразовать** (x -> x\*x)  
Последовательность .**Количество** (x -> x>2)  
Последовательность .**ОтсортироватьПо** (x->x.Поле)

```
uses Исполнители;  
  
begin  
  var посл := СлучайнаяПоследовательность(10);  
  посл.Вывести;  
  
  посл.Количество(значение -> Четное(значение)).Вывести;  
  
  var количество := 0;  
  foreach var значение in посл do  
    if Четное(значение) then  
      количество += 1;  
  количество.Вывести  
end.
```

```
3 10 1 6 10 10 1 4 8 3  
6  
6
```

# Цепочки методов

Методы можно объединять в цепочки

**Пример.** В последовательности целых **выбрать** значения  $x > 5$ , **преобразовать** эти значения в **квадраты** и **вывести** получившуюся последовательность



# Цепочки методов

Методы можно объединять в цепочки

**Пример.** В последовательности целых **выбрать** значения  $x > 5$ , **преобразовать** эти значения в **квадраты** и **вывести** получившуюся последовательность

Пишем алгоритм в точности так как сформулирована задача (!)

```
var послед := Последовательность (1, 5, 3, 7, 4, 9, 8) ;  
послед.Выбрать (x->x>5) .Преобразовать (x->x*x) .Вывести ;
```

# Цепочки методов

Методы можно объединять в цепочки

**Пример.** В последовательности целых **выбрать** значения  $x > 5$ , **преобразовать** эти значения в **квадраты** и **вывести** получившуюся последовательность

Пишем алгоритм в точности так как сформулирована задача (!)

```
var послед := Последовательность (1, 5, 3, 7, 4, 9, 8);  
послед.Выбрать (x->x>5) .Преобразовать (x->x*x) .Вывести;
```

**А так учат делать детей в школе и на ЕГЭ:**

```
var a,b: array [1..7] of integer;  
    i,k: integer;  
begin  
    a[1] := 1; a[2] := 5; a[3] := 3; a[4] := 7;  
    a[5] := 4; a[6] := 9; a[7] := 8;  
    for i:=1 to 7 do  
        if a[i]>5 then  
            begin  
                k := k + 1;  
                b[k] := a[i]*a[i]  
            end;  
        for i:=1 to k do  
            write (b[i], ' ');  
        end.  
end.
```



# Исполнитель Страны

- Исполнитель Страны является одновременно последовательностью и простой **базой данных**
- Страна(Название,Столица,Население,Континент)

```
uses Исполнители;  
  
begin  
  var количество := 0;  
  foreach var страна in Страны do  
    if страна.Название.НачинаетсяНа('А') then  
      begin  
        Вывести(страна);  
        количество += 1;  
      end;  
    количество.Вывести;  
  
  var страныА := Страны.Выбрать(страна -> страна.Название.НачинаетсяНа('А'));  
  страныА.ВывестиПострочно;  
  страныА.Количество.Вывести;  
end.
```

```
(Ангола, Луанда, 25830958, Африка)  
(Андорра, Андорра-ла-Велья, 76098, Европа)  
(Антигуа и Барбуда, Сент-Джонс, 92738, Америка)  
(Аргентина, Буэнос-Айрес, 43131966, Америка)  
(Армения, Ереван, 2998600, Азия)  
(Афганистан, Кабул, 33369945, Азия)
```

12

# Цепочечные запросы

- Некоторые запросы трудно реализовать в виде алгоритма
- Запросы позволяют отделить различные действия. При реализации обычного алгоритма эти действия смешиваются

```
uses Исполнители;
```

```
begin
```

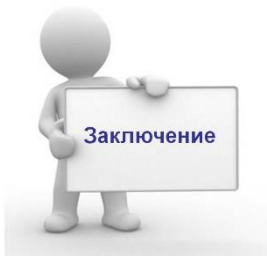
```
Страны.Выбрать(страна -> страна.Континент = 'Азия')  
  .ОтсортироватьПоУбыванию(страна -> страна.Население)  
  .Взять(10)  
  .ВывестиПострочно
```

```
end.
```

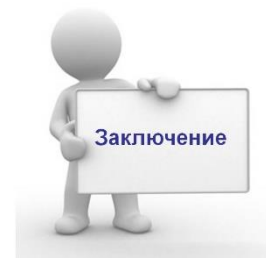
```
(Китай, 1379022000)  
(Индия, 1294186000)  
(Индонезия, 260581100)  
(Пакистан, 194521264)  
(Бангладеш, 160991563)  
(Япония, 126960000)  
(Филиппины, 102921200)  
(Вьетнам, 92700000)  
(Иран, 79003827)  
(Турция, 78741053)
```

# Апробация методики

- Воскресная компьютерная школа при мехмате ЮФУ (250 юных программистов каждый год)



# Заключение



- Возможность использования русских имён позволяет использовать PascalABC.NET для обучения программированию младших школьников
- Высокоуровневые исполнители в PascalABC.NET представляют новые базовые примитивы для построения алгоритмов
- Исполнители-последовательности и лямбды кардинально меняют и упрощают стиль решения многих стандартных задач
- Появляется возможность решать более содержательные и высокоуровневые задачи
- Возможно сочетание нескольких методик для достижения целей обучения
- Среда PascalABC.NET чрезвычайно гибка для обучения

# Спасибо за внимание!

```
uses Исполнители;  
  
begin  
  var s := 'Спасибо за внимание!';  
  s.ДляВсех(символ -> Вывести(символ))  
end.
```

# Лишний слайд

```
uses Окна,Исполнители;
```

```
begin
```

```
var СтраныА := Страны.Выбрать(страна -> страна.Название.НачинаетсяНа('А'));
```

```
var СтраныАзии := Страны.Выбрать(страна -> страна.Континент = 'Азия');
```

```
var МалСтраны := Страны.Выбрать(страна -> страна.Население < 100000);
```

```
var ОкноСписка := СоздатьОкноСписка;
```

```
ОкноСписка.Отобразить(Страны);
```

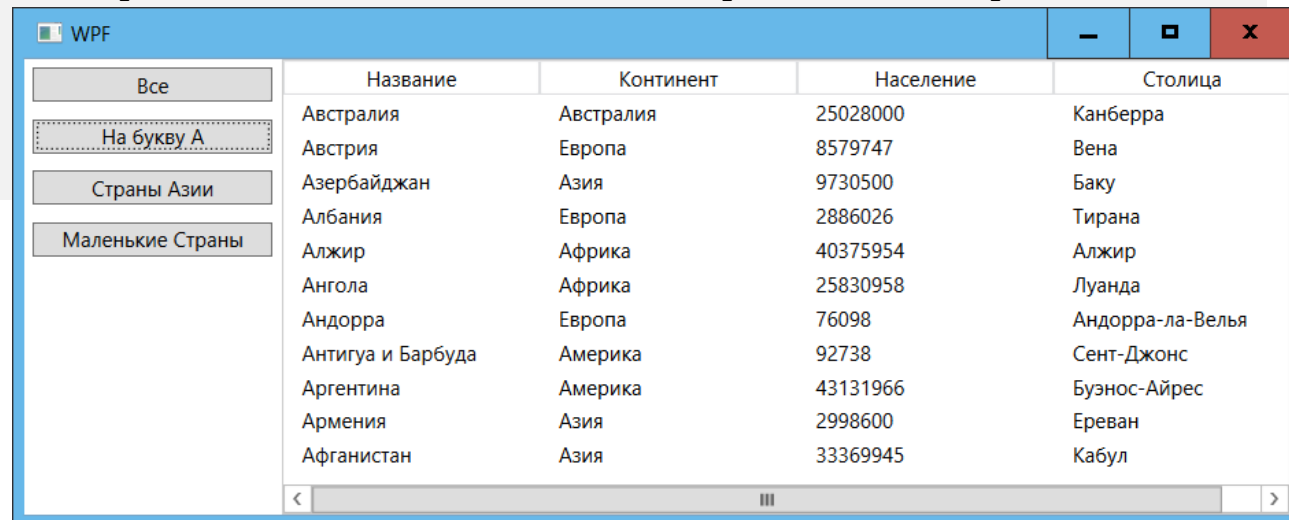
```
СоздатьКнопку('Все', () -> ОкноСписка.Отобразить(Страны));
```

```
СоздатьКнопку('На букву А', () -> ОкноСписка.Отобразить(СтраныА));
```

```
СоздатьКнопку('Страны Азии', () -> ОкноСписка.Отобразить(СтраныАзии));
```

```
СоздатьКнопку('Маленькие Страны', () -> ОкноСписка.Отобразить(МалСтраны));
```

```
end.
```



The screenshot shows a WPF application window titled "WPF". On the left side, there is a vertical panel with four buttons: "Все", "На букву А", "Страны Азии", and "Маленькие Страны". The "На букву А" button is currently selected, indicated by a dashed border. The main area of the window displays a table with the following columns: "Название", "Континент", "Население", and "Столица". The table contains 14 rows of data, listing countries and their corresponding continent, population, and capital.

Название	Континент	Население	Столица
Австралия	Австралия	25028000	Канберра
Австрия	Европа	8579747	Вена
Азербайджан	Азия	9730500	Баку
Албания	Европа	2886026	Тирана
Алжир	Африка	40375954	Алжир
Ангола	Африка	25830958	Луанда
Андорра	Европа	76098	Андорра-ла-Велья
Антигуа и Барбуда	Америка	92738	Сент-Джонс
Аргентина	Америка	43131966	Буэнос-Айрес
Армения	Азия	2998600	Ереван
Афганистан	Азия	33369945	Кабул