В центре системы находятся сетевые процессоры, имеющие архитектуру с распределенным интеллектом и осуществляющие автоматическое слежение за всеми составляющими «умного» здания. Такой метод работы определенно повышает надежность работы BMS, и сбой любого из процессоров никак не отразится на функциях контролируемой им системы жизнеобеспечения, так как его обязанности «подхватит» другой процессор.

Через центральный сервер и АРМ (автоматизированные рабочие места) операторов функциональных идет процесс архивирования данных, мониторинг работы систем жизнеобеспечения и интеллектуальном сооружении. Для обработки всей информации, идущей с любой системы жизнеобеспечения, в центре действует программное специальное обеспечение BMS.

Исследования большинства специалистов в этой области наталкивают нас на мысль о том, что, несмотря на масштабность и ответственность всего периода возведения здания, доля расходов на данный этап в совокупной стоимости затрат 40-летнего периода составляет только 11%. Расходы, касающиеся эксплуатации и ремонта, и будут являться себестоимостью владения, и лишь от них зависят арендные ставки и, собственно, объемы инвестиций. Более того, в связи с особенностями САиУЗ полностью владеть информацией о всем цикле жизнеспособности здания, срок эксплуатации дома до капитального ремонта возрастает до 50-55 лет.

1. Шишов О. В. Технические средства автоматизации и управления: учебное пособие. – Москва: ИНФРА-М, 2011 г. – 397 с.
2. Тесля Е.В. "Умный дом" своими руками. Строим интеллектуальную цифровую систему в своей квартире. — СПб., 2008. — 219 с.
3. Иванченко П. Технология LonWorks//Бюллетень "Автоматизация здания" 5/2007(10). — М.: ОАО "АСТ — Московский полиграфический дом", 2007. — 20 с.
4. Волков А.А. Информационное обеспечение в рамках концепции интеллектуального жилища//Жилищное строительство. -2001. -№8. -с. 4-5
5. Kharchenko L., Panova I., Babaeva E., Magomedova R., Gulyakin D., Pashtaev B. Russian syndromes, impeding the development of the innovation processes in education // Life Sci J 2014; 11 (9s): 184 -187, USA, (ISSN: 1097-8135). <http://www.lifesciencesite.com>. 35.
6. Гулякин Д.В. Современный вуз как среда, реплицирующая культуру общества // Научное обозрение. Серия 2: Гуманитарные науки. – 2012. – № 6. – С. 91-95.
7. Головин А. ВЫСТАВКА HI-TECH HOUSE: Бюллетень "Автоматизация зданий". Ноябрь 2006. №4.
8. Комаров Н.М., Жаров В.Г. Управление инженерными системами интеллектуального здания с использованием технологий информационного и инфографического моделирования // Сервис Plus. 2013. №2. С. 74-81
9. Федоров Е. "Город будущего" - энергоэффективность, безопасность и комфорт // Control engineering Россия. 2015. №2 (56). С. 22-26.
10. Шишов О. В. Современные технологии промышленной автоматизации : учеб. – Саранск: Изд-во Мордов. ун-та, 2009. – 276 с.

Пылькин А.Н., Соколова Ю.С., Дмитриева Т.А.

Выбор современного языка программирования для изучения базовых принципов программирования

*ФГБОУ ВО «Рязанский государственный радиотехнический университет им. В.Ф. Уткина»
(Россия, Рязань)*

doi: 10.18411/trnio-10-2021-45

Аннотация

Рассматривается ряд вопросов, связанных с выбором современного языка программирования для обучения, а также языковые средства современной версии языка Pascal PascalABC.NET, позволяющие писать короткий, понятный и эффективный код.

Ключевые слова: концепции программирования, языки программирования, PascalABC.NET, Python, обучение программированию.

Abstract

A number of issues related to the choice of a modern programming language for teaching are considered, as well as the language facilities of the modern version of the Pascal PascalABC.NET language, which allow you to write short, understandable and effective code.

Keywords: programming concepts, programming languages, PascalABC.NET, Python, teaching programming.

В последние годы интерес к IT-индустрии продолжает расти, а перед новичками и преподавателями возникает актуальный вопрос, какие технологии и языки программирования стоит изучить для успешного старта. Среди технологий программирования можно отметить такие технологии как структурное программирование, объектно-ориентированное программирование, а также функциональное и логическое программирование.

Объектно-ориентированное программирование является доминирующей концепцией программирования с 1990-х годов. Современная промышленная разработка программного обеспечения ведется на языках программирования, основанных именно на этой концепции. Поэтому знание объектно-ориентированного программирования является обязательным требованием работодателей к программистам. Объектно-ориентированное программирование является развитием двух предшествующих концепций: структурного и процедурного-модульного программирования. Поэтому для успешного освоения объектно-ориентированной концепции вначале нужно изучить эти предшествующие ей концепции [1].

Структурное программирование появилось в конце 1960 х годах с работ Э. Дейкстры [2, 3]. Основными положениями структурного программирования являются: использование только типовых управляющих структур следования, выбора и повторения, разработка программ с помощью нисходящего проектирования. Структурное программирование позволило сократить время на разработку и отладку программ, так как всегда можно было установить соответствие между этапом вычислительного процесса и фрагментом программы, его реализующего. Освоение технологии структурного программирования позволяет специалисту среднего уровня составлять программный код повышенной сложности, который в этом случае будет избыточным.

Увеличение сложности программных систем привело к трансформации структурного программирования в процедурное (процедурно-модульное) программирование в 1980 х годах. В структурном программировании не было четко установленных критериев деления программы на модули. В процедурном программировании был введен такой критерий модульности. Этот критерий был основан на понятии уровней абстракции [4, 5]. Каждая подпрограмма решала одну задачу на своем уровне абстракции с использованием подпрограмм непосредственно нижнего уровня.

Эволюция концепций программирования позволяла разрабатывать программные системы нового порядка сложности по сравнению с прежней, но делала инструменты разработки более трудными в освоении и использовании. Поэтому процесс обучения программированию также усложнялся.

Чтобы начать изучение объектно-ориентированного программирования, студент должен знать основы программирования и уметь представлять решение задачи в виде программы. При этом текст программы должен соответствовать концепциям структурного и процедурного программирования.

Перечислим базовые знания и умения в дальнейшем необходимые для освоения объектно-ориентированного программирования.

Основы программирования:

- типы данных;
- операции (в том числе математические);
- управляющие структуры;

- работа со сложными типами данных;
- ввод и вывод данных;
- отладка программы;
- оформление текста программы в соответствии с соглашениями языка программирования (например, стандарт PEP 8 написания программы на языке Python).

Структурное программирование:

- написание или преобразование текста программы в такой вид, чтобы она соответствовала концепции структурного программирования;
- нисходящее проектирование алгоритмов и программ.

Процедурное программирование:

- принципы деления программы на подпрограммы;
- способы обмена данными основной программы и подпрограмм;
- локальные и глобальные переменные;
- формальные и фактические параметры;
- создание библиотек модулей пользователя.

В настоящий момент существует большое число различных языков программирования, как классических, так и современных. Возникает вопрос, какой язык программирования выбрать для получения в этой области знаний и умений, перечисленных выше. Выбор усложняется тем, что рейтинги популярности языков программирования ежегодно меняются, а самих языков очень много, и каждый из них имеет свои особенности, сферы применения и решает свои задачи.

Basic (Бейсик). Был создан в 1964 году как интерактивный язык для студентов-непрограммистов, чтобы они после минимального обучения легко могли самостоятельно создавать программы [6]. Язык Basic является семейством языков высокоуровневого программирования, так как имеет большое число различных диалектов, ориентированных на интерпретацию программного кода (что приводит к снижению производительности машинных прогонов). В настоящий момент язык Basic из простого, почти примитивного языка превратился в самостоятельный язык Visual Basic, который используется для разработки прикладных программ, чаще всего для операционной системы Windows.

К недостаткам использования языка Basic можно отнести следующие: базовые разновидности языка не позволяют обучить структурному программированию [7], причиной этого является своеобразие управляющих конструкций и подпрограмм, значительно отличающаяся от синтаксиса в других языках программирования; язык Visual Basic предназначен не для начального обучения, а для решения других задач, в частности, для офисного программирования.

Pascal (Паскаль). Этот язык специально создан Н. Виртом в конце 1960 х – в начале 1970 х годов для обучения студентов структурному программированию [8]. В языке Pascal обязательно присутствует разделение программы на области (раздел описаний переменных, типов, констант, процедур и функций и др.), также особенностью является строгая типизация данных и однозначность толкования любой команды.

Как и Basic, язык имеет различные диалекты, в которые внесены некоторые изменения. Наиболее известной реализацией языка Pascal, обеспечившей широкое распространение и развитие языка, является Turbo Pascal фирмы Borland [9]. Дальнейшим развитием языка Pascal является язык Object Pascal, дополненный средствами объектно-ориентированного программирования. Язык Object Pascal используется в среде программирования Borland Delphi, в которой были внедрены значительные расширения языка [10].

В настоящее время в учебном процессе для изучения программирования с помощью языка Pascal широко используется свободно распространяемая среда разработки PascalABC, обладающая всем необходимым функционалом. Эта среда разработки функционирует на

платформе .NET. Язык программирования среды PascalABC.NET включает в себя не только классический язык Паскаль, но и большинство возможностей языка Object Pascal и все современные средства языков программирования, такие как классы, интерфейсы, лямбда-выражения, сборка мусора и многие другие. Данные особенности позволяют разделить обучение на необходимые этапы, указанные в данной статье, то есть сначала получить базовые знания по программированию, а потом расширить их знаниями по объектно-ориентированному программированию.

Недостатком языка Pascal является то, что он практически не применяется на практике промышленной разработки программ уровня сложности выше среднего. Но данный язык программирования до сих пор заслуженно считается одним из самых лучших языков для начального обучения программированию.

Python (Питон). Этот язык программирования подходит и для обучения, и для программирования сложных проектов. Язык Python является мощным высокоуровневым объектно-ориентированным языком программирования [11], который начал разрабатывать Г. Россум в конце 1980 х годов.

Именно этот язык считается в последние годы фаворитом для изучения основ программирования. Немногословный и понятный синтаксис, похожий на псевдокод и не усложненный громоздкими конструкциями, а также сильная динамическая типизация способствуют быстрому и безболезненному обучению новичков основам программирования. Кроме того, этот язык постоянно развивается, а сообщество вокруг Python – одно из самых сильных в мире IT.

Этот язык является кроссплатформенным, обладает последовательным синтаксисом, является модульным, легко масштабируется, благодаря чему исходный код написанных на Python программ легко читаем. Разрабатываемые программы содержат меньшее число строк кода по сравнению с другими языками программирования.

Язык Python поддерживает различные концепции программирования, в том числе структурное, процедурное, объектно-ориентированное и функциональное программирование. Для этого языка есть интерактивная консоль, в которой можно ввести команду и сразу увидеть ее результат. Кроме того, язык Python является интерпретируемым языком программирования, то есть для работы с ним подойдет любой текстовый редактор. Но для удобства работы можно использовать свободно распространяемую интегрированную среду разработки PyCharm Community Edition.

Язык Python – самый быстрорастущий язык программирования, который по рейтингу языков программирования от издания IEEE Spectrum с 2018 года уже находился на первом месте [12]. По другим рейтингам этот язык также находится на лидирующих позициях.

Основным недостатком использования языка Python для изучения основ программирования является то, что данный язык не формализован, в нем имеет место динамическая типизация, когда тип переменной определяется только во время исполнения. Это может спровоцировать возникновение целого ряда ошибок, которые начинающему программисту будет сложно выявить. Другим недостатком является то, что широкое использование этого языка программирования в нашей стране определяется недостаточностью периода его изучения и использования.

Тем не менее, язык программирования Python однозначно можно рекомендовать как на первом этапе обучения для получения начальных навыков программирования, так и на втором, при изучении технологии объектно-ориентированного программирования.

C (Си). Язык C разработал Д. Ричи в конце 1960 х – начале 1970 х годов [13]. Он является мощным, быстрым, кроссплатформенным языком программирования общего назначения. Язык C является первым языком, на котором написана операционная система. Язык C используется для разработки широкого спектра приложений от операционных систем, таких как Windows и iOS, до прикладного программного обеспечения для различных устройств и встраиваемых систем.

Этот язык также можно рассматривать в качестве языка для начинающих программистов. Язык С прост в освоении, в нем имеются различные возможности для написания сложных программ, он широко используется по всему миру. Знание языка С значительно облегчает в дальнейшем изучение других высокоуровневых языков, которые от него происходят (C++, Java, C#).

С точки зрения обучения, основным недостатком языка С является отсутствие полномасштабной реализации технологии процедурного программирования (нет понятия процедуры, своеобразное определение функции и др.)

Кроме того, выбор конкретного языка в профессиональной сфере зависит от цели деятельности. Например, можно отметить популярные серверные языки, такие как PHP, Java, Ruby. Из клиентских языков наиболее перспективен Java Script. Для разработки сложных высоконагруженных проектов и игровых приложений широко используется C++. Лидерами в разработке приложений для мобильных устройств считаются Java, Objective-C, Swift. Среди программ для разработки программ в сфере бизнеса, веб-сервисов, мобильных приложений, игр лидирующее положение занимают C# и Python.

Опыт преподавания авторами дисциплин программирования на первом и втором курсе в рамках укрупненного направления 09.00.00 – Информатика и вычислительная техника показывает, что наиболее приемлемыми языками для начального обучения технологиям программирования следует считать языки Pascal и Python.

Многие считают язык Pascal устаревшим, но это не так. Нужно отметить, что наравне с языком Python развиваются и совершенствуются другие языки программирования, включая PascalABC.NET [14, 15]. Рассмотрим более подробно возможности этого языка программирования и сравним их с возможностями языка Turbo Pascal и Python.

PascalABC.NET был создан около 15 лет назад в Южном федеральном университете [15]. Разработчики преследовали цель создания хорошего современного языка программирования для обучения школьников с мощной, хорошо русифицированной средой программирования, поддерживающей технологию IntelliSense, содержащей средства автоформатирования, встроенный отладчик и встроенный дизайнер форм, включающего в себя лучшее, что предлагают другие современные языки, такие как C#, Kotlin, Python, Haskell и другие. В качестве платформы реализации была выбрана платформа Microsoft.NET.

Синтаксически PascalABC.NET близок к языку Delphi, однако включает множество конструкций языка C# и содержит все современные языковые средства: классы, перегрузку операций, интерфейсы, обработку исключений, обобщенные классы и подпрограммы, сборку мусора, лямбда-выражения, средства параллельного программирования [15, 16]. Кроме того, он содержит собственные расширения языка Pascal, нацеленные на написание компактных, интуитивно понятных программ с использованием современных языковых возможностей.

Многое язык PascalABC.NET взял от языка Python. Это прежде всего легковесные языковые и библиотечные средства, обеспечивающие компактность и интуитивную понятность программ. Кроме того, часто код решения задачи, написанный на Python, фактически идентичен коду на PascalABC.NET.

Рассмотрим решение задачи (№ 14 из ЕГЭ по информатике) поиска количества повторений цифры 3 в значении выражения $4^{503} + 3 \cdot 4^{244} - 2 \cdot 4^{444} - 95$, записанного в системе счисления с основанием 4. В обоих случаях будет получено число 298.

Код на PascalABC.NET	Код на Python
##	
var N := 4bi**503 +	N = 4**503 +
3*4bi**244 -	3*4**244 -
2*4bi**444 - 95;	2*4**444 - 95
var k := 0;	k = 0
while N > 0 do	while N > 0:
begin	if N % 4 ==

<pre> if N mod 4 = 3 then k += 1; N := N div 4; end; print(k) </pre>	<pre> 3: k += 1 N = N // 4 print(k) </pre>
---	--

В рассмотренном примере для хранения значения N используется неограниченный целый тип, обозначаемый в PascalABC.NET идентификатором BigInteger.

Ещё один пример, демонстрирующий схожесть синтаксических конструкций этих языков программирования, также взят из сборников для подготовки к ЕГЭ по информатике (задача № 12): на вход программы для исполнителя «Редактор» поступила строка, состоящая из 107 букв X. Требуется определить, какая строка получится после выполнения следующего алгоритма (в результате должна получиться строка 'XX'):

```

ПОКА Нашлось (XXX) или Нашлось (ZYX) или Нашлось (ZXX)
    Заменить (XXX, ZZ)
    Заменить (ZYX, X)
    Заменить (ZXX, Y)
КОНЕЦ ПОКА

```

Код на PascalABC.NET

```

##
var s := 'X' * 107;
while ('XXX' in s) or ('ZYX' in s) or
('ZXX' in s) do
    begin
        s := s.replace('XXX', 'ZZ', 1);
        s := s.replace('ZYX', 'X', 1);
        s := s.replace('ZXX', 'Y', 1);
    end;
print(s)

```

Код на Python

```

s = 'X' * 107
while ('XXX' in s) or ('ZYX' in s) or
('ZXX' in s):
    s = s.replace('XXX', 'ZZ', 1)
    s = s.replace('ZYX', 'X', 1)
    s = s.replace('ZXX', 'Y', 1)
print(s)

```

PascalABC.NET, аналогично Python, позволяет использовать в программах идентификаторы, содержащие кириллические символы [15]. Так в предыдущем примере, реализовав подпрограммы Нашлось() и Заменить(), можно было оставить представленный исполнителю «Редактор» алгоритм фактически без изменений:

```

##
var s: string;
var Нашлось: string → boolean := v → s.Contains(v);
var Заменить: (string, string) → () := (v, w) →
    begin s := s.Replace(v, w, 1) end;
s := 'X'*107;

```

```

while Нашлось('XXX') or Нашлось('ZYX') or Нашлось('ZXX') do
begin
    Заменить('XXX', 'ZZ');
    Заменить('ZYX', 'X');
    Заменить('ZXX', 'Y')
end;
s.Print

```

Как можно заметить из примеров, в коде на PascalABC.NET нет привычных операторов `begin ... end`, разработчики заменили их на символы `##`. Кроме того, использование сочетания символов `###` включает в PascalABC.NET режим спортивного программирования, в котором операторы записываются в краткой форме: вместо `Print` можно писать `Pr`, вместо `Where – Wh`, вместо `Count – Cnt` и т.п.

Также из последнего листинга видно, что PascalABC.NET работает с кортежами, позволяет писать программы в функциональном стиле с использованием лямбда-выражений, которое легко распознается в тексте программы по характерному знаку операции $\rightarrow$ (произносится «переходит в ... »).

Использование кортежа в качестве выходных параметров у функции позволяет передавать в основную программу сразу несколько значений, например, с помощью функции можно сразу определить количество чётных и нечётных цифр в натуральном числе (пример не является учебным, он демонстрирует работу с кортежами):

```

##
function EvenOdd(n: integer): (integer, integer);
begin
    var (ev, od) := (0, 0);
    while n > 0 do
        begin
            var d := n mod 10;
            if d.IsEven then ev += 1 else od += 1;
            n := n div 10
        end;
    result := (ev, od)
end;

var num := ReadInteger('Введите число: ');
var (countEven, countOdd) := EvenOdd(num);
countEven.Println;
countOdd.Println

```

Модуль `School`, встроенный в современный PascalABC.NET, содержит реализацию алгоритмов, часто встречающихся в школьных задачах. Его использование делает текст программы яснее и короче, особенно для тех, кому не требуется углубляться в математические и технические детали этих алгоритмов. Так, с использованием модуля `School` и лямбда-выражений программа нахождения всех простых чисел из диапазона от a до b будет занимать три строки и иметь следующий синтаксис:

```

##
uses school;
var (a, b) := ReadInteger2('Введите два числа: ');
(a..b).Where(x → x.isPrime).Print

```


В PascalABC.NET программное решение задачи нахождения суммы чётных положительных чисел, не делящихся при этом на 8 или на 10, ввод которых заканчивается нулем, может быть записано одной строкой:

```
##
ReadSeqIntegerWhile(x → x<>0).Where(x → x.isEven and (x > 0) and
not(x.DivsAny(8, 10))).Sum.Print
```

Результатом работы программы при вводе последовательности:

13 -6 -2 4 35 64 22 70 -96 80 0 2 -6 44

будет число 26 ($4 + 22 = 26$).

Из приведенных примеров видно, что реализации некоторых функций имеют два формата – для вызова в виде функции и для записи в точечной нотации. Например, для вывода значения переменной s можно писать Println(s) или s.Println.

Но, несмотря на наличие мощной современной системы программирования на базе языка Pascal, в учебных заведениях на компьютерах с установленным PascalABC.NET продолжают писать код с синтаксисом для Turbo Pascal и преподавать устаревшие Free Pascal или Delphi. Многие не видят разницы между этими языками, полагая, что она минимальна или вообще отсутствует.

Следующий пример демонстрирует различие в кодах, написанных на Turbo Pascal, на PascalABC.NET и на Python, при решении задачи поиска минимума и его позиции в случайно сгенерированном массиве из 10 чисел, каждое из которых лежит в диапазоне от -100 до 100 (если минимумов несколько, определить номер первого).

Код на Turbo Pascal

```
var
  a: array[1..10] of integer;
  mn, imn, i: integer;
begin
  randomize;
  for i:=1 to 10 do a[i]:=random(101)-
random(101);
  for i:=1 to 10 do write(a[i]:4);
  writeln;
  mn := a[1];
  imn := 1;
  for i:=1 to 10 do
    if a[i] < mn then
      begin
        mn := a[i];
        imn := i;
      end;
  writeln(mn, ' ', imn)
end.
```

Код на PascalABC.NET

```
##
var a := ArrRandom(10, -100, 100);
a.Println;
a.Min.Print;
(a.IndexMin+1).Print;
```

Код на Python

```
from random import randint
```



```
a = [randint(-100, 100) for i in
range(10)]
print(a)
print(min(a))
print(a.index(min(a)) + 1)
```

Как показывает рассмотренный пример, коды на Turbo Pascal и PascalABC.NET сильно различаются. И несмотря на то, что в PascalABC.NET можно писать код в стиле старого Turbo Pascal, в нем заложены конструкции, присущие другим языкам программирования. Более того, он умеет понимать, форматировать и выполнять с определенными ограничениями программы, написанные на языке C# [15].

Если вспомнить про язык программирования Python [17, 18], который разрабатывался с целью повышения продуктивности самого программиста, нежели кода, который пишется, то PascalABC.NET также позволяет концентрироваться на сути алгоритма, а не на технических деталях благодаря своим встроенным библиотекам. Но, в отличие от языка Python, язык PascalABC.NET имеет статическую типизацию, что позволяет фиксировать множество ошибок на этапе компиляции и делает программу более безопасной. Кроме того, недостаток в скорости ограничивает область применения Python. Высокопроизводительное приложение на чистом Питоне написать не получится, потребуется прибегать к помощи других языков или использовать статически типизированные расширения. Поэтому, если вы планируете стать всесторонне развитым разработчиком, только знания Python недостаточно.

PascalABC.NET за пределами сферы образования практически не используется, но он имеет множество языковых возможностей, поддерживает множество стилей программирования, поскольку обобщает языковые и библиотечные возможности таких языков, как C#, Python, Kotlin, входящих в TOP-10 рейтинга востребованных языков программирования 2021 [19].

Не стоит гнаться за изучением современных языков программирования, поскольку, как показывает практика, на смену одним кодам приходят другие. Гораздо важнее обучать базовым принципам программирования, а этого можно добиться, используя на начальном этапе изучения технологий программирования язык PascalABC.NET. Так, этот язык взят за основу в курсе «Основы программирования» у студентов 1 курса мехмата ЮФУ по направлению «Фундаментальная информатика и информационные технологии» [15]. Объясняется это тем, что PascalABC.NET является мультипарадигменным языком, позволяющим программировать в структурном, объектно-ориентированном и функциональном стилях, а теоретическая и практическая база, полученная при его изучении, позволяет эффективно освоить современные технологии программирования и в очень короткий срок изучать впоследствии такие языки как Python, C++, C#, Java, Haskell, Ruby и др.

PascalABC.NET является удачным решением для начального знакомства с современными технологиями программирования еще и потому, что это программное обеспечение является свободно распространяемой отечественной разработкой с интегрированным электронным задачником Programming Taskbook [20], который предназначен для обучения программированию и содержит учебные задания, охватывающие все основные разделы базового курса программирования: от скалярных типов и управляющих операторов до сложных структур данных и рекурсивных алгоритмов.

1. Dmitrieva T.A., Prutskov A.V., Pylkin A.N. Two-level study of object-oriented programming by university students / Современные информационные технологии и ИТ-образование № 1, том 15, 2019.
2. Dijkstra E.W. Structured programming. In J.N. Buxton, B. Randell (eds.) Software Engineering Techniques, NATO Scientific Affairs Division, Brussels, Belgium, 1969 (pp. 84-88).
3. Dahl O.J., Dijkstra E.W., Hoare, C.A.R. Structured programming. Aca-demic Press, 1972.

4. Dijkstra E.W. The Structure of the "THE"-Multiprogramming System. In Communications of the ACM, 11 (5), May 1968 (pp. 341-346).
5. Liskov B.H. A Design Methodology for Reliable Software Systems. In Proc. of AFIPS'72 (Fall, part I), 1972 (pp. 191-199).
6. Basic. A Manual for BASIC, the elementary algebraic language designed for use with the Dartmouth Time Sharing System. Dartmouth College, 1964. Available at: http://www.bitsavers.org/pdf/dartmouth/BASIC_Oct64.pdf (дата обращения: 04.09.2021).
7. Bogardus Cortez, M. History of Basic Computer Programming Languages: C Language, BASIC and Beyond. In EdTech magazine, June 2017. Available at: <https://edtechmagazine.com/higher/article/2017/06/history-programming-languages-c-language-basic-and-beyond> (дата обращения: 10.09.2021).
8. Niklaus Wirth. The Programming Language Pascal. In Acta Informatica, 1, 1971 (pp. 35-63).
9. Chapman, M.R. In Search of Stupidity: Over 20 Years of High-Tech Marketing Disasters, 2nd ed. Apress, 2006.
10. Gibson, S. Borland and Microsoft Enter the Object-Oriented Pascal Ring. In Infoworld, 11 (9), 1989 (p. 28).
11. Srinath K.R. Python – The Fastest Growing Programming Language. In International Research Journal of Engineering and Technology (IRJET), 4 (12), 2017 (pp. 354-357).
12. Cass, S. The Top Programming Languages 2018. Python stays on top, and Assembly enters the Top Ten. In IEEE Spectrum, July 2018. Available at: <https://spectrum.ieee.org/at-work/innovation/the-2018-top-programming-languages> (дата обращения: 15.09.2021).
13. Ritchie, D. The Development of the C language. In HOPL-II The 2nd ACM SIGPLAN conference on History of Programming Languages, Cambridge, Massachusetts, USA, 1993 (pp. 201-208).
14. https://pascalabcnet.github.io/mydoc_for_EGE.html (дата обращения 22.09.2021).
15. Осипов А.В. PascalABC.NET: Введение в современное программирование. – Ростов-на-Дону, 2019 – 572 с.
16. Осипов А.В. PascalABC.NET: выбор школьника. Часть 3. /А.В. Осипов. – Ростов-на-Дону, 2020 – 146 с.
17. Лутц, Марк. Изучаем Python, том 1, 5-е изд.: Пер. с англ. – СПб.: ООО "Диалектика", 2019. – 832 с.
18. Лутц, Марк. Изучаем Python, том 2, 5-е изд.: Пер. с англ. – СПб.: ООО "Диалектика", 2020. – 720 с.
19. <https://habr.com/ru/post/543346/> (дата обращения 20.05.2021).
20. <http://www.ptaskbook.com/ru/> (дата обращения 28.09.2021).

Смирнов С.В.

**Проектирование тематического слоя электронной карты при помощи графа
качественных признаков объектов**

*ИПУ РАН
(Россия, Москва)*

doi: 10.18411/trnio-10-2021-46

Аннотация

В докладе описано проектирование тематического слоя электронной карты специализированной геоинформационной системы (ГИС) при помощи мозаичной иерархической модели изобразительных средств. Рассмотрен граф понятийно - содержательных признаков. Рассмотрена схема графа формирования изобразительных средств для проектирования знаков. Обосновано применение способа квадратомического дерева для ГИС.

Ключевые слова: геоинформационной системы (ГИС), граф, модель, тематический слой, электронная карта, картография, квадратомическое дерево, иерархическая структура, графическая информация.

Abstract

The report describes the design of the thematic layer of the electronic map of a specialized geoinformation system (GIS) using a mosaic hierarchical model of visual means. The graph of conceptual and substantive features is considered. The scheme of the graph of the formation of visual means for the design of signs is considered. The application of the quadratic tree method for GIS is justified.

Keywords: geographic information system (GIS), graph, model, thematic layer, electronic map, cartography, quadratic tree, hierarchical structure, graphic information.