

Школа LXF

Спонсор рубрики
PingWin Software!
Созданная в мае 2009 года компания занимается поддержкой свободных продуктов, сообществ их разработчиков, пользователей и внедренцев.
www.pingwinsoft.ru

Обмен опытом и передовые идеи по использованию свободного ПО в образовании

Исполнители и алгоритмы

Говорят, в английском языке все слова взяты из C++. **Александр Казанцев** идет в обратном направлении и разбирается, как можно применить в программировании язык русский.



Наш эксперт

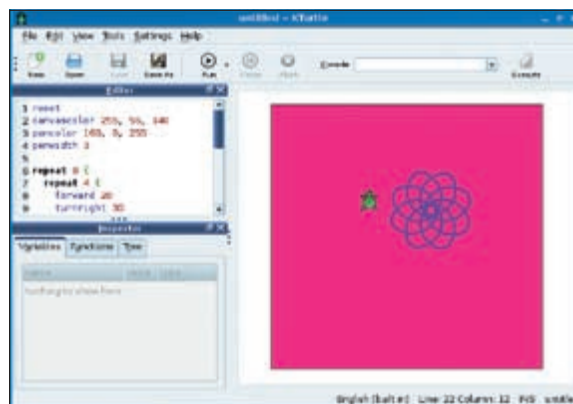
Александр Казанцев

К.т.н., доцент кафедры Информатики Глазовского государственного пединститута, руководитель проекта EduMandriva, автор локализаций (и разработок) для Mandriva, Klavaro, Gambas2 и LXDE.

Информатика — наука сложная. С этим трудно поспорить. Особенно нелегко тем людям, у которых есть проблемы с логическим построением каких-либо задач. Логика лежит в основе множества составных частей информатики, а особенно — программирования. Последнее непосредственно связано с прикладной частью информатики. Получается замкнутый круг, из которого нужно искать выход.

С другой стороны, само программирование таит в себе скрытые проблемы — информатика из академической науки, сродни математике, превращается во всеобъемлющий инструмент для применения в других областях — и вот мы видим, что она востребована во многих возрастных группах и предметах: на русском языке, географии, физике, рисовании, музыке, в детском саду и начальной школе.

Вы можете возразить: а как это связано с проблемами программирования? Да очень просто. Если вернуться к определениям, то программа — это алгоритм, записанный на каком-либо языке. Этот язык по большей степени непонятен целевым группам (ради интереса подсуньте программу на C++ учителю музы-



➤ Красивый цветочек — но если забыть пакет локализации KDE, то черепашка будет понимать только английский язык.

ки или четырехлетнему ребенку). С другой стороны, алгоритм как порядок действий является достаточно понятной вещью, и человек может составить его после непродолжительного этапа даже не обучения, а пояснения порядка действий (опять же получается каламбур: порядок действий для составления порядка действий). Но здесь мы упираемся в проблему, упомянутую на первом этапе — логику. Теперь вам все понятно?

Судя по вашим округлившимся глазам, уследить за полетом моей мысли смогли не все, поэтому не буду загружать вас «академическими рассказами» (поверьте, это я умею), а перейду непосредственно к делу.

Исполнители

Итак, что поможет нам развить логическое мышление, познакомиться с понятием алгоритма и даже обучить программированию? Ответ достаточно прозаичен и известен многим — это исполнители (роботы) и псевдокод. Первые выполняют роль посредников между пониманием команд, отдаваемых машине, и получаемым

Исполнитель и псевдокод = ?

Обратившись к Википедии, вы можете легко найти определения и первого, и второго. Из них можно заключить, что псевдокод — компактный способ описания алгоритмов, использующий ключевые слова языков программирования, но опускающий несущественные подробности и специфический синтаксис. Псевдокод содержит основные конструкции языков программирования и иногда даже повторяет их — тогда говорят о Паскаль- или Бейсик-подобном псевдокоде и т.п. — или пишется на естественном языке. Одним из пер-

вых языков программирования и псевдокодом одновременно был Алгол (сокращение от английских algorithmic — алгоритмический и language — язык) — дедушка всех императивных языков. Самым известным псевдокодом в России является разработанный академиком А.П.Ершовым в середине 1980-х годов школьный алгоритмический язык, который изначально был призван решить проблему безмашинного обучения, но после был применен А.Г.Кушниренко для управления исполнителями.

результатом; второй помогает постичь саму суть алгоритма. Так как и первые, и вторые появились не прямо сейчас и не сегодня, то в настоящий момент существуют и исполнители, и псевдокоды. И их использование при обучении не только рекомендуется, но и требуется (например, не забегая вперед, скажем что один из вариантов псевдокода входит в задания Единого государственного экзамена по информатике). Опять же, исполнители не всегда будут работать под управлением псевдокода.

И вновь (наверное, я уже замучил вас затянувшейся интригой и постоянным подыманием проблем?): что делать при использовании свободного ПО и Linux в частности? Что может предложить нам мир свободы и доступности кодов? Давайте рассмотрим все по порядку.

Исполнители имеют одно общее свойство – у них есть определенный набор команд, который они понимают. Они бывают двух видов: исполнители «обстановки» и исполнители «величин». Давайте рассмотрим, какие исполнители мы можем найти «на просторах» открытого и свободного ПО.

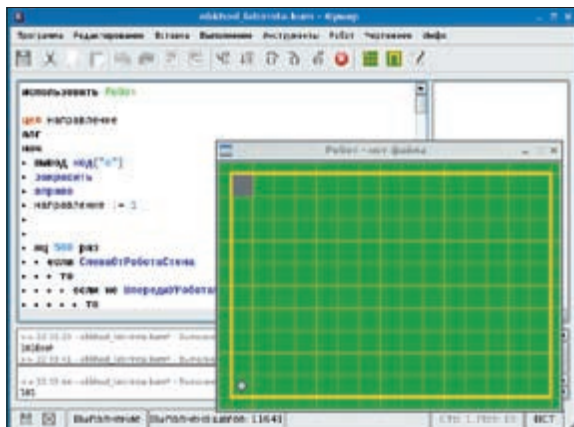
Черепашка и все, все, все

Самым известным исполнителем является «черепашка» (turtle). Она появилась в результате разработки педагогического программного средства обучения детей алгоритмизации психолога Сеймура Пейперта [Seymour Papert] – языка программирования Лого. Иными словами, черепашка является исполнителем алгоритмов, записанных на языке Лого.

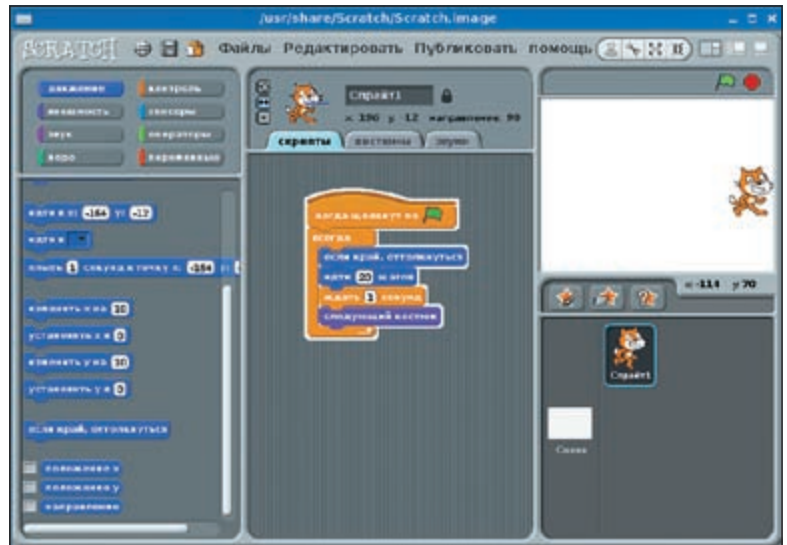
В Linux также можно найти «черепашку» – приложение *KTurtle* является частью пакета образовательных приложений *KDE-Edu* и существует как для третьей, так и для новой, четвертой версии этого рабочего стола. Даже если у вас установлен дистрибутив, который основан не на KDE (например, Ubuntu), вы можете подключить репозиторий, содержащий KDE, и доустановить пакеты. Не забудьте также добавить пакеты локализации для KDE, иначе придется изучать язык оригинала.

Для более продвинутых пользователей (а также тех, у кого проблемы с приложениями KDE или кто не хочет тянуть «лишние» библиотеки) существует библиотека «черепашьей графики» для Python с понятным названием *turtle* (<http://docs.python.org/library/turtle.html>).

Черепашка оказала большое влияние на развитие другого проекта – *КуМир*. Исполнители Робот и Чертежник – по сути, Черепашка, разделенная на две составляющие. Хотя *КуМир* и был создан во второй половине 80-х годов, в настоящий мо-



» Робот будет долго искать выход из этого «лабиринта».

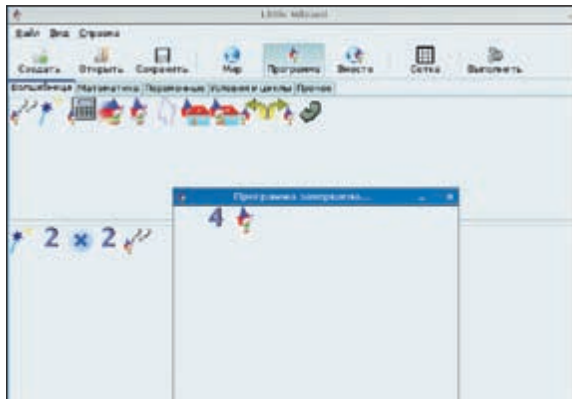


» Даже такая маленькая программа может заставить кота шагнуть от стенки к стенке.

мент идет активная работа над его продолжателем, который использует библиотеку *Qt* и поддерживает операционные системы GNU/Linux и Windows. Разработка ведется в НИИСИ РАН под руководством М.А. Ройтберга. С сайта проекта (<http://lpm.org.ru/kumir/>) вы можете скачать последнюю версию пакетов (а также версию для разработчиков, которая уже содержит более двух исполнителей) под известные дистрибутивы Linux, или исходные коды для самостоятельной сборки.

В отличие от «черепашки», исполнители *КуМир*'а используют алгоритмический язык, который является частью заданий Единого государственного экзамена, что позволяет безболезненно и углубленно вписать их в программу обучения.

Еще более продвинутыми являются исполнители, программный код к которым написан на другом учебном диалекте «большого» языка программирования, SmallTalk. Мы говорим об учебных проектах, которые основаны на языке *Squeak*. Пожалуй, самым известным исполнителем, который написан на этом языке, является *Scratch* (<http://scratch.mit.edu>) – проект создания программ на диалекте языка Лого (то есть можно применять навыки, полученные при работе с Черепашкой), но без использования непосредственной записи алгоритмов. В *Scratch* программа набирается из «кирпичиков», как в конструкторе (авторы проекта не скрывают, что черпали идеи в конструкторе Лего и специализированном синтаксисе Лего-Лого), где строительными блоками являются команды действий исполнителя, работа с переменными, отслеживание событий (как во взрослых объектных языках программирования), циклы, условия, переходы и т.д. В *Scratch* можно создавать фильмы, играть с различными объектами, видеоизменять их, перемещать по экрану, устанавливать формы взаимодействия между объектами. То есть можно сказать, что это объектно-ориентированная среда, в которой блоки программ собираются из разноцветных кирпичиков-команд. Поэтому вы можете использовать *Scratch* как в пропедевтических курсах в младших классах, так и на уроках программирования в старших – возрастная категория достаточно обширна. *Scratch* также будет хорошей заменой роликам на основе проприетарных Flash и Action Script – создавать мультфильмы, размещать их на сайте проекта в Интернете и показывать родителям, знакомым и друзьям здесь можно за пару щелчков мыши.



› $2 \times 2 = 4$, не так ли?

Вы можете найти последнюю версию *Scratch* в учебных редакциях ваших любимых дистрибутивов, или скачать пакеты (к сожалению, только в формате `.deb`) с сайта проекта http://info.scratch.mit.edu/Linux_installer. Там же можно прочитать, как установить *Scratch* из исходных текстов с помощью специального скрипта на Perl.

Что есть еще?

Можно упомянуть и другие исполнители, которые не столь известны, но заслуживают вашего внимания. В первую очередь это Маленькая Волшебница – *Little Wizard*. Она позволяет создавать программы для исполнителя с помощью кубиков-команд с рисунками. То есть воспользоваться этой программой могут самые маленькие «программисты», которые только учатся читать и писать. С помощью *Миров* можно построить новый дом, провести волшебницу по созданному лабиринту или дорожке. Причем алгоритмы могут быть достаточно сложными: есть даже циклы, позволяющие избежать дублирования похожих участков кода. Два режима – Числа и Алфавит – помогут составлять примеры и выводить текст. А поддержка ветвлений еще больше усложнит программный продукт.

«Позволяет создавать программы с помощью кубиков-команд.»

Little Wizard можно найти в репозиториях вашего дистрибутива или загрузить на сайте проекта – <http://littlewizard.sourceforge.net/>.

Кто-то скажет, мол, все это – для малышни, а есть ли в Linux что-то более сложное? Если коротко, то да. Одним из примеров «сложного» исполнителя является *Robocode* (<http://robocode.sourceforge.net/>). Сказать по правде, это даже не исполнитель, а поле боя для программистов. Каждый его участник пишет на Java код, управляющий небольшим танком-роботом, который сражается с другими такими же танками. Основная цель – уничтожить все остальные танки на поле. Без компромиссов. А при чем тут обучение, и как согласуется война и школа? Ничего страшного – в *Robocode* нет крови, нет людей и нет политики. В игре есть взрывы; но если вы заботитесь о психическом здоровье школьников, их можно легко отключить. У танков большие возможности: они могут двигаться по игровому полю, стрелять, узнавать местоположение друг друга, врезаться в стены и в других роботов, выяснять, где находятся летящие снаряды, и использовать всю мощь языка Java. Естественно, какой-либо простой стратегии для выигрыша нет. Существует огромное количество различных тактик, у каждой из которых есть свои преимущества и недостатки.

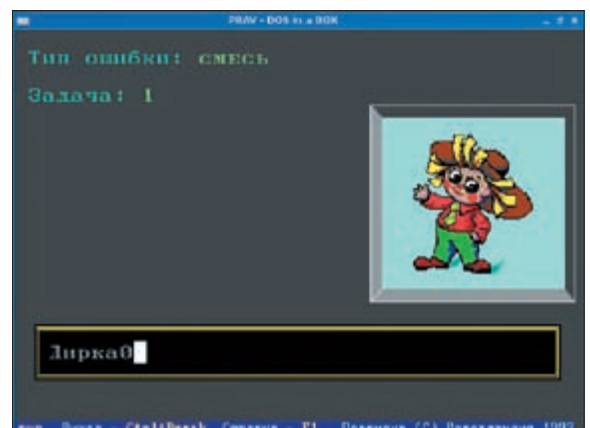
А как быть с проверенным ПО?

Уже слышу хор возмущений: «Подсунули тут неизвестные продукты!» Да, в настоящий момент, не считая *КуМир*'а, под Linux не существует родных версий столь известных вещей, как *Роботландия*, *ЛогоМиры* или *Паркетчик*. Что делать, если у вас есть копии этих продуктов, причем лицензионные? Сразу предупредим: все описанные далее программы имеют или закрытый код, или проприетарную лицензию, но опять же аналогов им на настоящий момент не существует (надеемся, эта статья послужит для кого-нибудь посылком к действию).

Ничего страшного. Начнем с самого известного (к сожалению, в настоящий момент – во все более узких кругах) пакета программных продуктов «Роботландия».

Он разработан еще под DOS, поэтому его выполнение в среде *Wine* невозможно. Но это не означает, что с ним нельзя работать в Linux (на вопрос о том, надо ли использовать такой старый продукт, я обычно отсылаю узнать возраст *Vi* и *Bash*).

Запуск *Роботландии* отлично реализуется с помощью DOS-эмулятора *dosemu* и установленной в нем свободной версии DOS – FreeDOS. Вы также можете найти их в репозиториях вашего дистрибутива. Установив эмулятор, вы получите прекрасно работающую DOS-среду, в которой можно запустить *Роботландию* (а также старого доброго *КуМир*'а, кроссплатформенная версия ко-



› Битва в самом разгаре. Кто победит?

› Знакомая «Правилка», но под Linux.

того не доросла до уровня DOS-редакции, *Кенгуренка*, и другие хорошие проверенные обучающие продукты, которые, к сожалению, почил в веках). Затем вы сможете проверить работу исполнителей *Роботландии*, и наконец-то познакомиться с предметом обожания всех маленьких компьютерщиков начала 90-х – Петей Куком и его друзьями (и не только с исполнителями), проживающими на сайте <http://www.botik.ru/~robot>.

Там же вы найдете «роботландский» аналог «черепашки» – *Кукарачу*, которая идет в среде *Wine* и (к сожалению) не имеет ничего общего со свободным ПО.

Паркетчик (его вы встретите в учебниках информатики за авторством А.Г. Гейна) представляет собой исполнителя, который, в отличие от «черепашки», не рисует, а выкладывает плитку (нет, ремонт в ванной и на кухне он сделать не поможет). Управление

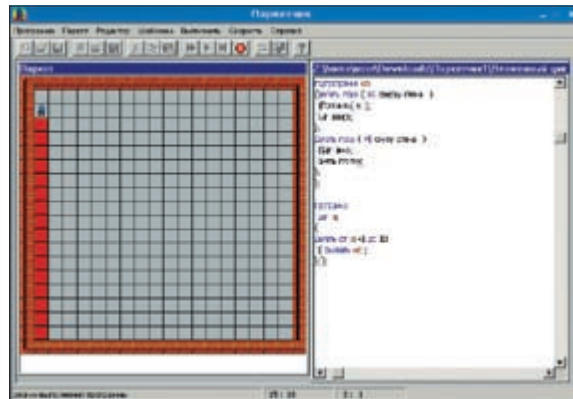
исполнителем ведется на основе Паскаль-подобного псевдокода на русском языке. Этот исполнитель также прекрасно работает в среде *Wine*, и можно скачать его бесплатно распространяемую версию с http://www.prosv.ru/info.aspx?ob_no=19088, где есть и более сложные исполнители – машины Поста и Тьюринга.

ЛогоМиры также прекрасно чувствуют себя в *Wine*. И даже более того – разработчики обещают скоро выпустить новую версию продукта (под кодовым именем «Пломбир»), которая будет считать Linux своим родным домом. Поэтому повсеместное внедрение и использование Linux будет как раз тем стимулом, который подстегнет разработчиков создавать, портировать и адаптировать свои продукты под Linux, а возможно, вступать в дружный круг создателей свободного ПО, так как идеи, витающие в нем, как раз совпадают с идеями образования. **LXF**


Полезные ссылки

Чтобы не заставлять вас рыскать по просторам Интернета, мы сделали подборку сведений по исполнителям, которая, надеемся, будет актуальна в ближайшее время.

- » Руководство по *KTurtle* (если вы не поставите себе KDE) – <http://docs.kde.org/stable/ru/kdeedu/kturtle/index.html>
- » Алгоритмизация и основы программирования на базе *KTurtle*: Учебное пособие – http://window.edu.ru/window/catalog?p_rid=58396
- » *Scratch* на Летописях.py – <http://letopisi.ru/index.php/Скретч>
- » Учиться со *Scratch* – <http://setilab.ru/scratch/>
- » Программируем на Лого и Python – <http://freecode.pspo.perm.ru/115/>
- » *LittleWizard* – <http://freecode.pspo.perm.ru/236/littleWizard/index.html>



» «Ложит» плитку, а потом перекладывает снова. Ничего не напоминает?..



Академическая программа для учебных заведений

Mandriva Linux

Mandriva.Ru предоставляет учебным заведениям лицензию, дающую право на неограниченное по числу рабочих станций использование дистрибутива Mandriva Linux на всех компьютерах в образовательном учреждении, всех компьютерах преподавателей и всех компьютерах учащихся, в том числе и домашних.

По этой программе учебное заведение получает:

- ★ свежие версии дистрибутива Mandriva Linux (дважды в год)
- ★ доступ к обновлениям системы
- ★ техническую поддержку

Комплект поставки:
Mandriva Linux Powerpack 2009.1 Spring — 32- и 64-битные версии (2 DVD), а также печатное руководство
★ Mandriva Free 2009.1 Spring
★ Mandriva One 2009.1 Spring
★ Репозиторий Mandriva 2009 — бинарные пакеты для платформы x86 (4 DVD)
★ EduMandriva (1 DVD) — дополнительное ПО для образования
★ Академическая лицензия

Цена: 30 тысяч рублей

Тел.: (812) 309-06-86, (499) 271-49-55
info@mandriva.ru, www.mandriva.ru