

ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON В ШКОЛЬНОМ КУРСЕ ИНФОРМАТИКИ

А. И. Лапо

*Лицей им. Ф. Э. Дзержинского Белорусского государственного университета,
Минск, Беларусь, lapo@lyceum.by*

В статье рассматриваются содержательные, методические и дидактические особенности изучения языка программирования Python на базовом уровне изучения учебного предмета «Информатика» для учреждений образования, реализующих образовательные программы общего среднего образования. Акцент сделан на использование системно-задачного подхода.

Ключевые слова: учебный предмет информатика; язык программирования; Python; методика преподавания информатики.

PYTHON PROGRAMMING LANGUAGE IN SCHOOL COMPUTER SCIENCE COURSE

A. I. Lapo

*F. E. Dzerzhinsky Lyceum of Belarusian State University
Minsk, Belarus, lapo@lyceum.by*

The article discusses the substantive, methodical and didactic features of studying the Python programming language at the basic level of studying the subject "Computer Science" for educational institutions implementing educational programs of general secondary education. The emphasis is on the use of a system-task approach.

Keywords: subject computer science; programming language; Python; methods of teaching computer science.

1. Введение

С 2024 года в Беларуси начался процесс переиздания учебных пособий по учебному предмету «Информатика» для учреждений образования, реализующих образовательные программы общего среднего образования. Уже переиздано учебное пособие для 6 класса и идет издательская работа по подготовке учебного пособия для 7 класса. В ближайшие два года будут переизданы учебные пособия для базовой школы, а затем и для средней школы.

Наиболее масштабные изменения в учебных пособиях затронут главы, относящиеся к содержательной линии «Алгоритмизация и программирования». Все учебные пособия по информатике 2008–2022 гг. использовали язык программирования Pascal для изучения основ алгоритмизации и программирования. Однако современность требует нового взгляда на этот раздел школьного курса информатики. Поэтому в учебных пособиях язык программирования Pascal постепенно заменяется на язык программирования Python.

В учебной программе по учебному предмету «Информатика» не указан конкретный язык программирования, на основе которого учащиеся знакомятся с основами алгоритмизации и программирования. Большинство школьных учителей опираются на тот язык программирования, который предложен в учебном пособии.

2. Критерии отбора языка программирования для школьного курса информатики

Наиболее существенные критерии в выборе языка программирования приведены в [1]. Вот некоторые из них:

- в языке должны быть реализованы те алгоритмические конструкции и типы данных, которые прописаны в учебной программе;
- язык должен быть доступен для понимания учащимися, а среда должна позволять учащимся создавать и отлаживать реально работающие программы;
- использование языка в различных интеллектуальных соревнованиях учащихся;
- законность использования программного обеспечения;
- наличие и доступность учебных пособий, соответствующих учебной программе;
- уровень квалификации учителя;
- популярность языка программирования в различных профессиональных рейтингах.

Использование на олимпиадах по информатике только трех языков программирования: Pascal, C++ и Python, существенно сужает возможность выбора языка для обучения основам алгоритмизации и программирования на базовом уровне в учреждениях образования, реализующих образовательные программы общего среднего образования.

Язык программирования C++ достаточно сложен для начального обучения программирования, поскольку в нем отсутствуют простые библио-

теки для работы с графикой и библиотеки для работы с формальными компьютерными исполнителями. Он рекомендуется для изучения на профильном уровне.

Язык программирования Python на протяжении пяти лет занимает первую строчку в рейтинге Tiobe [2] (Приложение 1). И хотя язык программирования Pascal вернулся в первую десятку рейтинга, его рейтинг (2,26%) гораздо меньше, чем у языка Python (25,98%). Синтаксис языка программирования Python не сложнее чем у Pascal, для него разработано огромное количество различных библиотек, в том числе и для работы с искусственным интеллектом.

Все эти факторы являются несомненными плюсами в пользу выбора языка Python в качестве языка для начального знакомства с основами алгоритмизации и программирования.

3. Учебные пособия по информатике

Учебные пособия по информатике, которые издаются в Беларуси с 2017 года, имеют двухполосную верстку. Полосы выделяются цветом, в зависимости от содержания размещенного там материала.

На полосе белого цвета размещается основной теоретический материал. Примеры, поясняющие основной материал, размещаются на полосе голубого цвета. На полосе голубого цвета могут быть вставки на салатном фоне, где размещается дополнительный материал: интересные факты, исторические сведения, материалы о практическом применении изучаемой темы в различных сферах жизни человека в Беларуси.

Изначальная идея такой верстки была в том, чтобы на белом фоне разместить такую информацию, которая с течением времени не сильно устаревает. Поскольку, в большинстве случаев, примеры в учебных пособиях по информатике ориентированы на конкретное программное обеспечение, то их вынесли в отдельную полосу голубого цвета. Поэтому при изменении программного обеспечения, используемого для изучения конкретной темы, материал на белом фоне меняется незначительно, а основные изменения происходят с примерами, которые размещены на голубом фоне. В табл. 1 приведены страницы из учебных пособий «Информатика 6» 2018 [3] и 2024 [4] годов издания. Текст, приведенный на белом фоне, мало изменился при переиздании пособия. Однако примеры программ, приведенных на голубом фоне написаны на разных языках программирования. В пособии 2018 года пример написан языке программирования Pascal, в пособии 2024 года на языке Python.

Страницы учебных пособий «Информатика 6»

Пособие 2018 года	Пособие 2024 года
<p>Пример 19.9. Программа построения изображения из трех одинаковых цифр.</p> <pre> uses Drawman; begin Field(7,4); //первая тройка слева ToPoint(1,3); PenDown; OnVector(1,0); OnVector(-1,-1); OnVector(1,0); OnVector(-1,-1); PenUp; //средняя тройка ToPoint(3,3); PenDown; OnVector(1,0); OnVector(-1,-1); OnVector(1,0); OnVector(-1,-1); PenUp; //третья тройка справа ToPoint(5,3); PenDown; OnVector(1,0); OnVector(-1,-1); OnVector(1,0); OnVector(-1,-1); PenUp; end. </pre> <p>мент программы и использовать его нужное число раз.</p> <p>Так, в примере 19.9 требуется записать программу для построения изображения, состоящего из трех цифр «3».</p> <p>Мы видим, что программу построения изображения можно составить на основе программы из примера 19.8. Изображение первой цифры начинается от верхней точки слева, ее координаты (1, 3). Координаты такой же точки для второй цифры (3, 3), для третьей цифры (5, 3).</p> <p>Таким образом, для создания изображения из трех цифр «3» нужно скопировать в тексте программы примера 19.8 следующий фрагмент:</p> <pre> ToPoint(1,3); PenDown; OnVector(1,0); OnVector(-1,-1); OnVector(1,0); OnVector(-1,-1); PenUp; </pre> <p>Затем следует вставить скопированный фрагмент нужное количество раз и внести изменения.</p>	<p>В некоторых изображениях часто повторяются одинаковые фрагменты. Для создания программ построения таких изображений можно скопировать повторяющийся фрагмент программы и использовать его нужное число раз так, как это делали в примере 19.8 для построения елки.</p> <p>Пример 19.11. Написать программу для построения трех одинаковых цифр «3» красного, зеленого и синего цвета.</p> <p>Мы видим, что программу построения этого изображения можно составить на основе программы из примера 19.8. Изображение первой цифры начинается от верхней точки слева, ее координаты (–50, 60). Координаты такой же точки для второй цифры (0, 60), для третьей – (50, 60).</p> <p>Таким образом, для создания изображения из трех цифр «3» в тексте программы из примера 19.10 нужно скопировать фрагмент</p> <pre> turtle.penup() turtle.setpos(-50,60) turtle.pendown() turtle.forward(30) </pre> <p>Значения параметров команд Черепашки можно записывать выражениями. Например, вместо команды <code>turtle.forward(42)</code> можно записать команду <code>turtle.forward(30*1.41)</code>.</p> <p>Пример 19.11. Программа построения изображения из трех одинаковых цифр разного цвета.</p> <pre> import turtle turtle.shape('turtle') turtle.penup() #первая тройка turtle.penup() turtle.color('red') turtle.setpos(-50,60) turtle.forward(30) turtle.right(135) turtle.forward(42) turtle.left(135) turtle.forward(30) turtle.right(135) turtle.forward(42) turtle.left(135) #вторая тройка turtle.penup() turtle.color('green') turtle.setpos(0,60) turtle.pendown() turtle.forward(30) turtle.right(135) turtle.forward(42) turtle.left(135) #третья тройка turtle.penup() turtle.color('blue') turtle.setpos(50,60) turtle.pendown() turtle.forward(30) turtle.right(135) turtle.forward(42) turtle.left(135) </pre>

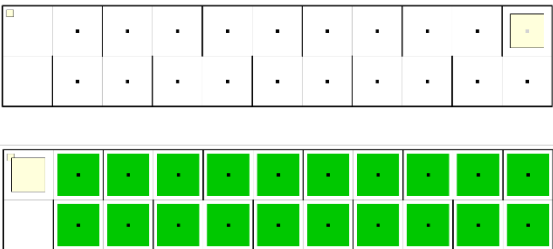
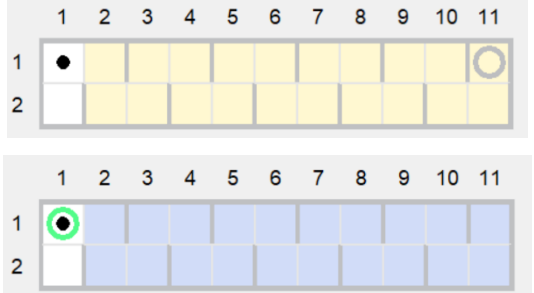
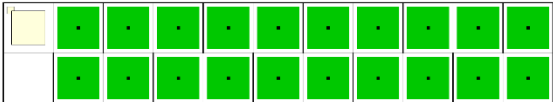
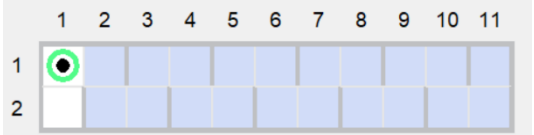
Изучение основ алгоритмизации и программирования в 6 классе является пропедевтическим и основывается на работе с формальным компьютерным исполнителем. При использовании языка программирования Pascal рассматривался исполнитель Чертежник. При переходе на язык программирования Python используется компьютерный исполнитель Черепашка. Невзирая на разных исполнителей и разные языки программирования, примеры, приведенные в пособиях, практически не изменились.

В 7 классе изучение основных алгоритмических конструкций происходит с помощью компьютерного исполнителя Робот. Исполнитель Робот входит в стандартную установку среды программирования PascalABC.Net. Для языка Python нет стандартного Робота, тем не менее в интернете можно найти большое количество библиотек с реализацией исполнителя Робот на языке программирования Python. Для переиздания учебного пособия для 7 класса [5] авторы взяли за основу библиотеку, разработанную в МФТИ [6]. Предоставленные там материалы распространяются под лицензией CC-BY-SA, что позволило внести в библиотеку необходимые коррективы. В частности, были изменены некоторые команды исполнителя Робот. Вместо длинных и сложных словосочетаний типа «`wall_is_on_the_left()`» были введены более короткие «`wall_left()`», поскольку учащиеся в 7 классе не обладают высокой скоростью набора и не все изучают английский язык. Для исполнителя Робот должны быть созданы обстановки, в которых он может перемещаться. Все обстановки помещаются в задачник. Задачник для учебного пособия был полностью написан с нуля. В него вошли все примеры и упражнения,

содержащиеся в пособии. Задачник разрабатывался так, чтобы обстановки Робота совпадали с теми, которые были представлены в учебном пособии 2017 года издания. В табл. 2 представлены начальные обстановки и результаты работы исполнителя Робот к одной и той же задаче «Закрасить клетки поля *Робота* по образцу», но реализованные на разных языках программирования.

Таблица 2

Начальная и конечная обстановки Робота

Pascal	Python
	
	

Такой подход к разработке задачника для Робота позволил сохранить большую часть материалов учебного пособия, которые были размещены на белом фоне.

В 8 классе учащиеся изучают основные алгоритмические конструкции в языке программирования. При переиздании учебного пособия планируется сохранить те задачи, которые были в учебном пособии для 8 класса [7]. Поэтому большинство изменений, которые затронут учебное пособие, опять будут на голубом фоне. Та система задач, которая сейчас используется в учебном пособии, охватывает все структурные элементы учебной программы и позволяет формировать алгоритмическую компетентность учащихся. Аналогичный подход будет поддерживаться и при переиздании учебного пособия для 9 класса.

4. Дидактические особенности изучения языка программирования Python

В базовой школе (6–9 класс) а рамках содержательной линии «Основы алгоритмизации и программирования» изучаются следующие темы:

- Алгоритмы и исполнители (6 класс, 8 часов);
- Основные алгоритмические конструкции (7 класс, 12 часов);
- Основы алгоритмизации и программирования (8 класс, 14 часов);
- Алгоритмы обработки строковых величин (9 класс, 8 часов).

Одним из основных дидактических принципов является *принцип научности*. В применении к изучению основ алгоритмизации и программирования это означает, что обучение строится на достоверных знаниях о языке программирования и логике построения алгоритмов. Это необходимо для формирования алгоритмических компетенций, позволяющих развить у учащихся алгоритмическую компетентность. На пропедевтическом уровне (6–7 классы) к таким компетенциям можно отнести [8]: алгоритмические компетенции, связанные с выполнением готового алгоритма и поиска, и исправления ошибок в алгоритме, алгоритмические компетенции разработки и выполнения простейших алгоритмов с использованием цикла, ветвления, вспомогательного алгоритма. На базовом уровне (8–9 класс): алгоритмические компетенции, обеспечивающие запись алгоритма на языке программирования и отладки программы, а также алгоритмические компетенции разработки алгоритмов и записи и выполнения программ для обработки числовых и символьных данных с использованием основных алгоритмических конструкций. Для реализации данного принципа язык программирования Python подходит как же, как и другие языки программирования.

Язык Python имеет низкий порог входа и простой синтаксис. Отсутствие фигурных скобок `{ }` или конструкций типа **begin end**; и использование отступов приучает к написанию аккуратного, структурированного кода с самого начала. Это способствует реализации принципа *доступности*. И язык Python имеет здесь явные преимущества перед другими языками программирования.

Реализации принципа *наглядности* способствует наличие большого количества разнообразных библиотек, разработанных для Python, а том числе и графических. Библиотека черепашей графики turtle позволяет не только познакомиться с компьютерным исполнителем, но и реализовать простейшие графические построения и даже анимацию.

Принцип *связи теории с практикой* позволяет показать программирование не только как инструмент для решения реальных задач, но ориентировать учащихся на выбор профессий в сфере ИТ, поскольку Python обладает самым высоким рейтингом среди языков программирования.

Другие дидактические принципы (систематичность и последовательность, доступность, индивидуализация и дифференциация, развивающий и воспитывающий характер обучения) не выявляют преимущество Python перед другими языками программирования в процессе обучения по содержательной линии «Основы алгоритмизации и программирования». Реализация этих принципов обучения не изменится при переходе от языка программирования Pascal к языку программирования Python.

5. Методические особенности изучения языка программирования Python

Переход на новый язык программирования в явном виде не затронет учащихся, поскольку будет происходить постепенно. Те учащиеся, которые начинали обучение основам алгоритмизации и программирования на языке Pascal, продолжат обучение на нем. Те, кто начал изучать Python, не столкнутся (если только сами не захотят), с языком Pascal.

Учебные пособия ориентированы на системно-задачный подход к обучению программирования [9]. Это означает, что главным объектом является задача, вся совокупность задач, размещенных в учебных пособиях, составляет систему. Задачи подобраны в соответствии с принципами *открытости, сложности и нелинейности*. Эти принципы определены в работах С. М. Окулова ([10], [11]). Построенная система задач должна содержать наглядные и понятные примеры. Согласно теории Балла [12] система задач должна содержать *индивидуальные задачи и родовые*, каждой из которых соответствует некоторый класс индивидуальных задач. В учебных пособиях родовые задачи разобраны как примеры, а индивидуальные находятся в разделе «Упражнения».

Остановимся на некоторых методических принципах обучения программированию. Эти принципы реализуются вне зависимости от языка.

Принцип *поэтапного формирования умственных действий* означает, что освоение основ программирования происходит по этапам:

- ознакомление (что такое цикл?);
- репродукция (повтори, поясни, прочитай код);
- применение (измени код);
- конструирование (используй фрагменты кода из разных программ);
- творчество (напиши свою программу).

Здесь можно привести аналогию с уроками литературы, на которых учащихся знакомят с поэзией. Они читают, объясняют, анализируют стихи. Могут подбирать рифмы к отдельным словам, но никто не требует от учащихся написания стихов. Это могут сделать только те, кто обладает определенным талантом к стихотворному творчеству. Также и написание полностью, с нуля программы относится к творчеству и не требуется в учебной программе (за исключением повышенного уровня изучения предмета).

Принцип *индивидуализации и дифференциации* предполагает учет уровня подготовки, темпа обучения, интересов каждого учащегося. Для этого разным учащимся предоставляются задания разного уровня сложности (базовый, продвинутый, исследовательский). В соответствии с этим принципом с учащимися разбираются не все, имеющиеся в учебных пособиях, примеры, а только те, которые учащиеся смогут освоить как родовые задачи.

Индивидуальные задачи затем подбираются под освоенные родовые. Сегодня чат-боты с искусственным интеллектом способны генерировать большое количество однотипных заданий за несколько секунд.

Принцип *обратной связи* позволяет обучить учащихся самоконтролю. Этому способствует анализ ошибок в программе, проверка кода с помощью тестирующих систем (например, в интернете), самооценка, исправление ошибок в программе с помощью отладки.

6. Заключение











Информатика в настоящее время развивается очень быстро. Это касается информатики и как раздела науки, и как учебного предмета в базовой школе. Быстрые изменения требуют постоянного обновления содержания учебных пособий по учебному предмету «Информатика». Вместе с тем сегодня в информатике как науке уже сформировано фундаментальное ядро, элементы которого лежат в основе учебной программы. Содержание учебного предмета «Информатика» последовательно раскрывается в процессе обучения по содержательным линиям (разделам). Содержательная линия «Основы алгоритмизации и программирования» не может быть реализована без конкретного языка программирования. Изучение языка программирования не должно стать самоцелью. Язык программирования должен рассматривать как инструмент для решения задач поскольку он является средством для формирования алгоритмической компетентности учащихся. Изучение языка программирования на основе системно-задачного подхода позволяет ориентироваться в первую очередь на задачу, и только потом на выбор конкретного языка, с помощью которого она может быть решена. Задачи, рассматриваемые в базовом курсе информатики должны быть такими, чтобы их решения на разных языках программирования были почти одинаковыми. Переход на язык программирования Python – это выбор нового инструмента для решения стандартных задач. При этом цели и задачи изучения учебного предмета «Информатика», а также формируемые компетенции останутся неизменными.

Библиографические ссылки

1. Лапо А. И. Язык программирования при изучении учебного предмета «Информатика» // Непрерывное образование в области естественных наук и математики: организация, методология, технологии: сборник материалов V научно-практической конференции с международным участием 4-5 июля 2024 года / Витебский областной институт развития образования; редкол.: А. Н. Деревяго (отв. ред.) [и др.]. Витебск: Государственное учреждение дополнительного образования взрослых «Витебский областной институт развития образования», 2024. С. 170–176.

2. TIOBE Index. // URL: <https://www.tiobe.com/tiobe-index/> (дата обращения: 18.09.2025).
3. *Макарова Н. П., Лапо А. И., Войтехович Е. Н.* Информатика: Учеб. пособие для 6 класса учреждений общ. сред. образования с рус. яз. обучения. Минск : Народная асвета, 2018.
4. Информатика. Учебное пособие для 6 класса учреждений образования, реализующих образовательные программы общего среднего образования, с русским языком обучения и воспитания. 2-е издание, пересмотренное и дополненное. / В. М. Котов [и др.]. Минск : Народная асвета, 2024.
5. *Котов В. М., Лапо А. И., Войтехович Е. Н.* Информатика: Учеб. пособие для 7 класса учреждений общ. сред. образования с рус. яз. обучения. Минск : Народная асвета, 2017.
6. Практика: Робот. URL: http://judge.mipt.ru/mipt_cs_on_python3/labs/lab2.html (дата обращения: 18.09.2025).
7. Информатика: Учеб. пособие для 8 класса учреждений общ. сред. образования с рус. яз. обучения / В. М. Котов [и др.]. Минск : Народная асвета, 2018.
8. *Лапо А. И.* Компетентностный подход в обучении школьников программированию // Педагогика информатики. 2020. № 1. URL: https://pcs.bsu.by/2020_1/n2.html (дата обращения: 18.09.2025).
9. *Лапо А. И.* Методика построения систем задач по программированию в школьном курсе информатики // Вес. БДПУ. Сер. 3, Фізика. Матэматыка. Інфарматыка. Біялогія. Геаграфія. 2014. № 4. С. 50–56.
10. *Окулов С. М.* Информатика: развитие интеллекта школьников. М. : БИНОМ. Лаборатория знаний, 2008.
11. *Окулов С. М.* Развитие интеллекта школьника как принцип организации синергетической среды обучения информатике: автореф. дис. ... д-ра пед. наук : 13.00.02 / МГПУ, 2004. 56 с.
12. *Балл Г. А.* Теория учебных задач: Психолого-педагогический аспект. М.: Педагогика, 1990.

Рейтинг языков программирования Tiobe за сентябрь 2025

Sep 2025	Sep 2024	Change	Programming Language		Ratings	Change
1	1			Python	25.98%	+5.81%
2	2			C++	8.80%	-1.94%
3	4	▲		C	8.65%	-0.24%
4	3	▼		Java	8.35%	-1.09%
5	5			C#	6.38%	+0.30%
6	6			JavaScript	3.22%	-0.70%
7	7			Visual Basic	2.84%	+0.14%
8	8			Go	2.32%	-0.03%
9	11	▲		Delphi/Object Pascal	2.26%	+0.49%
10	27	▲		Perl	2.03%	+1.33%