

DOI: 10.32517/2221-1993-2023-22-5-57-61

Г. В. Гаркавенко*Воронежский государственный университет, г. Воронеж, Россия***Ю. С. Бакулина, И. Е. Сильвестров***Воронежский государственный педагогический университет, г. Воронеж, Россия*

МОДЕЛИРОВАНИЕ ТЕКСТОВОГО ИНТЕРФЕЙСА СРЕДСТВАМИ ЯЗЫКА PYTHON

Аннотация

В статье рассматриваются вопросы, связанные с изучением в курсе информатики языка программирования Python и улучшением понимания обучающимися такой структуры языка программирования, как массив или список. Для достижения этой цели используется один содержательный пример — моделирование в текстовом интерфейсе работы табло электронных часов. Сначала разрабатываются шаблоны цифр для представления их на табло, при этом можно проработать темы, связанные с изучением матриц или списков строк. Далее поставленная задача реализуется средствами языка программирования Python, при этом для создания и обработки вывода цифр используются функции. Также приводятся идеи аналогичных задач. Статья может быть полезна учителям информатики, студентам педагогических вузов и всем изучающим программирование на языке Python.

Ключевые слова: обучение программированию, Python, структура данных «список», списки строк, функции в Python, содержательная задача.

1. Введение

В жизни часто приходится сталкиваться с данными, представленными в виде списка или таблицы. Это, например, список фамилий учащихся в классном журнале или таблица квадратов чисел на форзаце учебника по алгебре; список фильмов, идущих в кинотеатре, и расписание их сеансов; зал кинотеатра, в котором в соответствии с билетом надо отыскать сначала ряд, указанный в билете, а затем место в этом ряду. В математике такие структуры называют векторами и матрицами, а в про-

граммировании их принято называть массивами, списками, двумерными массивами.

При изучении раздела «Алгоритмы и программирование» в школьном курсе информатики все чаще используется язык программирования Python [5, 8, 10]. Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нем программ. Язык является полностью объектно-ориентированным в том плане, что всё в нем

Контактная информация

Гаркавенко Галина Валериевна, канд. физ.-мат. наук, доцент кафедры математического обеспечения ЭВМ, Воронежский государственный университет, г. Воронеж, Россия; *адрес:* 394018, Россия, г. Воронеж, Университетская пл., д. 1; *e-mail:* g.garkavenko@mail.ru

Бакулина Юнона Сергеевна, студентка физико-математического факультета, направление подготовки 09.03.03 «Прикладная информатика», профиль «Прикладная информатика в образовании», Воронежский государственный педагогический университет, г. Воронеж, Россия; *адрес:* 394043, Россия, г. Воронеж, ул. Ленина, д. 86; *e-mail:* nona.bakulina.1@mail.ru

Сильвестров Илья Евгеньевич, студент физико-математического факультета, направление подготовки 09.03.03 «Прикладная информатика», профиль «Прикладная информатика в образовании», Воронежский государственный педагогический университет, г. Воронеж, Россия; *адрес:* 394043, Россия, г. Воронеж, ул. Ленина, д. 86; *e-mail:* ilyasilvestrov@yandex.ru

G. V. Garkavenko

Voronezh State University, Voronezh, Russia

Yu. S. Bakulina, I. E. Silvestrov

Voronezh State Pedagogical University, Voronezh, Russia

MODELING A TEXT INTERFACE USING PYTHON LANGUAGE

Abstract

The article discusses issues related to studying the Python programming language in an informatics course and improving students' understanding of such a programming language structure as an array or list. To achieve this goal, one meaningful example is used, that is modeling the operation of an electronic clock display in a text interface. First, patterns of numbers are developed to represent them on the scoreboard, and topics related to the study of matrices or lists of strings can be better developed. Next, the task is implemented using the Python programming language, while functions are used to create and process the output of numbers. Ideas for similar problems are also given. The article may be useful to informatics teachers, students of pedagogical universities and all readers who are interested in programming in Python.

Keywords: programming training, Python, "list" data structure, lists of strings, functions in Python, meaningful task.

является объектами. Необычной особенностью языка является выделение блоков кода пробельными отступами. Синтаксис ядра языка минималистичен, за счет чего он, наверно, и нравится изучающим программирование школьникам.

В учебнике по информатике для IX класса К. Ю. Полякова, Е. А. Еремина изучение программирования ведется с использованием языка Python, в этом учебнике в главе 4 рассматривается работа со строками, одномерными массивами и матрицами (двумерными массивами) [12, 14]. В учебном пособии «Информатика. 8–9 классы. Начала программирования на языке Python. Дополнительные главы к учебникам», написанном коллективом авторов под руководством Л. Л. Босовой, также предложены для изучения строковый тип данных, массивы и функции [2]. В своей статье [1] Л. Л. Босова пишет о необходимости изменений в подходах к изучению информатики в школе. И, конечно, одной из главных задач, как, впрочем, при изучении любого предмета, является задача мотивировать обучающихся, заинтересовать их изучением информатики [3]. Сделать это можно также, предложив для решения содержательные [9] или исследовательские задачи [4–6].

Цели данной статьи:

- дать обучающимся наглядное представление об устройстве одномерных массивов и матриц посредством моделирования элементов текстового интерфейса, путем позиционирования элементов, создающих текстовый интерфейс;
- развить умение работать со структурами данных в языке Python;
- стимулировать интерес к изучению программирования [3, 9].

2. Разработка модели интерфейса электронных часов

Для изучения таких элементов языка программирования, как строки, массив (список), функция, и для пропедевтики изучения матриц обучающимся предлагается реализовать следующую содержательную задачу:

Написать программу, имитирующую работу электронных часов, которые показывают время вида «часы : минуты : секунды». Например: 09:20:15. Программа должна работать в текстовом режиме.

Решение поставленной задачи разобьем на два этапа:

- 1) моделирование интерфейса электронных часов в тетради;
- 2) программная реализация смоделированного интерфейса.

Итак, опишем первый этап работы над задачей, который предназначен для отработки представления о позиции элемента в матрице.

1. Сначала разработаем на листе в клетку шаблоны для будущих цифр. Пример электронных часов приведен на рисунке 1 (фото с сайта https://kitsanshop.ru/elektronnye_chasy_tablo_kvarc_2_szel_ind).

Учтем при проектировании модели циферблата электронных часов, что, например, 2 часа дня будет соответствовать 14 часам. Все цифры должны быть



Рис. 1. Пример электронных часов

размещены внутри прямоугольников одинакового размера, состоящих из клеток, и часть клеток будет закрашена.

Пример шаблона цифр размером 6×4 представлен на рисунке 2.

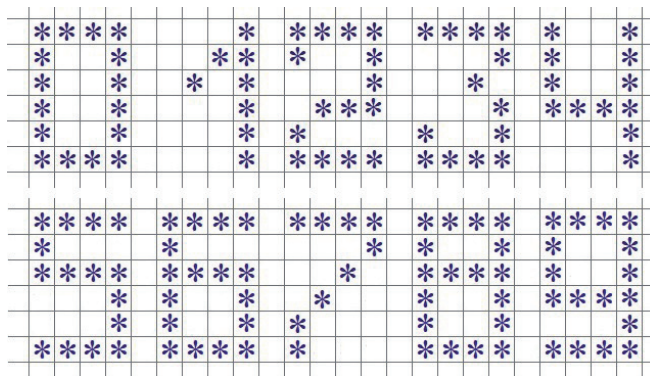


Рис. 2. Образец шаблонов цифр

Обучающиеся могут предложить другой шаблон. Например, в своей программе мы использовали шаблон размером 6×5, т. е. 6 рядов по 5 элементов в ряду.

2. Если посмотреть на электронные часы, изображенные на рисунке 1, то можно заметить, что часы представляются двумя цифрами: даже если в данный момент 8 часов утра, то на экране это будет представляться как 08, это же касается минут и секунд. Затем следует разработать шаблон для отделения часов от минут и секунд с помощью двоеточия, размер такого шаблона должен соответствовать размеру шаблонов цифр — так будет удобнее выводить время на экран.

3. Учтем, что мы работаем в текстовом режиме. В этом случае мы не можем вывести на экран одну цифру, а затем вернуться на ту строку, с которой начинали печать, и напечатать следующую цифру. Надо все цифры начинать печатать на экране одновременно. Посмотрите внимательно на верхнюю строку на рисунке 2. Мы видим пример того, что представляет собой строка, в которой находятся первые строки шаблонов для одновременной печати нескольких цифр.

4. Теперь выберем структуру для хранения таких шаблонов. Шаблоны цифр представляют собой прямоугольники-таблицы и напоминают ряды кресел в кинотеатре, часть из которых заполнена. Такие структуры аналогичны матрицам, но мы будем их рассматривать как списки строк. Каждая строка будет представлять собой ряд ячеек шаблона, например, на рисунке 2 первую строку шаблона размером 6×4 для цифры 0 можно записать в виде «****», вторую — как «* *» (звездочка, два пробела, звездочка), и так все шесть строк шаблона.

3. Разработка программы, реализующей интерфейс электронных часов в текстовом формате на языке программирования Python

Сначала кратко напомним используемые в программе элементы языка Python.

Строки в данном языке программирования являются неизменяемыми объектами, а поскольку символьного типа данных как такового нет, то строки можно записывать как в кавычках, так и в апострофах, т. е. в программе можно будет писать «***» или '***'.

В Python **списки** — это упорядоченные изменяемые коллекции объектов произвольных типов. Создание списка можно осуществлять путем указания его элементов в квадратных скобках через запятую. Например: `my_list = [1, 2, 3, 4, 5]`. В качестве элементов списка может использоваться любой тип данных, в том числе строки, как в проектируемой нами программе.

Списки в Python могут быть использованы для многих задач, таких как хранение данных, обработка и анализ данных, управление потоком выполнения программы и т. д. Важно понимать, как работать со списками, чтобы максимально эффективно использовать их в своей программе.

Для улучшения структуры программы будем использовать **функции** в Python, что позволит сделать программу более читаемой. Описание функции имеет вид, представленный на рисунке 3.

```
def ИмяФункции(параметры):
    операторы, реализующие действия функции
    return возвращаемое значение
```

Рис. 3. Общий вид функции в Python

1. Опишем необходимые для создания интерфейса электронных часов действия:

- задание шаблонов цифр и разделителя (двоеточия);
- определение того, шаблон какой цифры надо выводить на экран;
- получение времени;
- выделение цифр, задающих часы, минуты и секунды;
- вывод на экран построчно времени.

2. Шаблон для каждой цифры реализуем в виде функции, возвращающей список строк, и назовем ее соответствующим образом. Например, приведенные на рисунках 4 и 5 функции `F0()` и `F7()` задают по шаблону 6×5 списки для цифр 0 и 7 разной реализации:

```
def F0():
    return [ " *** ",
            " *  * ",
            " *  * ",
            " *  * ",
            " *  * ",
            " *** " ]
```

Рис. 4. Пример шаблона для цифры 0

```
def F7():
    a = [] * 6
    a[0] = "*****"
    a[1] = " *  * "
    a[2] = " *  * "
    a[3] = " *  * "
    a[4] = " *  * "
    a[5] = " *  * "
    return a
```

Рис. 5. Пример шаблона для цифры 7

Аналогично надо реализовать все цифры и еще функцию `Ddot()` для вывода двоеточия. Чтобы обращаться к элементам списка строк, нужно указывать номер строки в квадратных скобках. Например, `a[0]` будет обращаться к первой строке списка `a`, а `a[1]` — ко второй строке.

3. Далее реализуем функцию для выбора цифры, которую надо напечатать. Ее фрагмент приведен на рисунке 6.

```
def FF(p):
    if p==0:
        return F0()
    if p==1:
        return F1()
```

Рис. 6. Выбор нужной цифры

4. В основной части программы будем определять текущее время и выводить его на экран в текстовом режиме.

Основная часть программы работает по следующему алгоритму:

- 1) Описываются функции для печати каждой цифры. Эти функции возвращают список строк, представляющих данную цифру.
- 2) Определяется функция `FF(p)`, которая в качестве параметра получает цифру и возвращает массив строк, представляющий цифру, соответствующую числу `p`.
- 3) Импортируются необходимые библиотеки: `time` — для получения системного времени и `os` — для работы с операционной системой.
- 4) В цикле программа получает текущее время с помощью функции `time.strftime('%H:%M:%S').split(':')`. Эта функция возвращает список из часов, минут и секунд.
- 5) Часы, минуты, секунды представляют собой однозначное или двузначное число, которое, чтобы вывести на экран с помощью подготовленных для этого функций, надо сначала разбить на цифры. А затем, например, для вывода каждой цифры, задающей часы, формируются два списка, каждый из которых отвечает за свою цифру.
- 6) Цикл `for` выводит на экран часы построчно, т. е. сначала выводятся элементы всех списков с индексом 0, затем 1 и т. д. С помощью функции `time.sleep(1)` осуществляется задержка времени между обновлениями экрана часов. Если не использовать эту функцию, то экран будет обновляться чаще и несколько раз выводить одинаковое время. В программе задан еще цикл с условием `while`, который ограничивает работу программы 10 выводами на экран компьютера. При написании программы обучающиеся сами могут задавать это количество или убрать данный цикл.

Замечание. Если вы работаете в среде программирования, которая ввод и вывод делает с помощью терминала, например `geany`, то можно добавить команду очистки окна с помощью функции `os.system('clear')` для Linux и `os.system('CLS')` для Windows. Программу можно также запустить в командной строке. Пример работы программы представлен на рисунке 7.

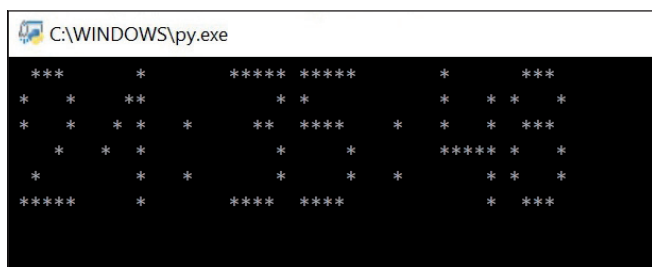


Рис. 7. Вид запуска программы

Основная программа представлена на рисунке 8.

```
import time
import os
k=0
while k<10:
    t=time.strftime('%H:%M:%S').split(':')
    h1=FF(int(t[0])/10)
    h2=FF(int(t[0])%10)
    m_1=FF(int(t[1])/10)
    m_2=FF(int(t[1])%10)
    s_1=FF(int(t[2])/10)
    s_2=FF(int(t[2])%10)
    dot=Ddot()
    print(end='')
    for i in range(0,6):
        print(h1[i],h2[i], dot[i],
              m_1[i],m_2[i],dot[i],s_1[i],s_2[i])
    print()
    time.sleep(1)
    k+=1
    os.system('CLS')
```

Рис. 8. Основная программа

4. Заключение

В данной статье авторы представили содержательную задачу по программированию [9] и ее решение на языке программирования Python. Полученная программа представляет собой пример использования средств Python для создания интерфейса электронных часов с помощью символьной графики на экране.

Изучая работу данной программы, обучающиеся лучше освоят принцип работы таких неотъемлемых и важных элементов языка Python, как списки, строки, функции, познакомятся с некоторыми функциями из таких полезных библиотек, как time и os.

Используя данный пример, на уроке можно организовать групповую работу школьников, предложив каждому создать шаблон какой-то одной цифры или двоеточия, а затем собрать всё в единый программный код.

Программа может быть полезна для изучения основ программирования на Python и создания простых приложений для терминала.

Приведем еще идеи для создания программ, моделирующих работу электронных табло.

Задача 1.

Написать программу, реализующую интерфейс электронного табло-календаря, показывающего дату в формате «число.месяц.год» (например, 01.09.2023) или как на рисунке 9.

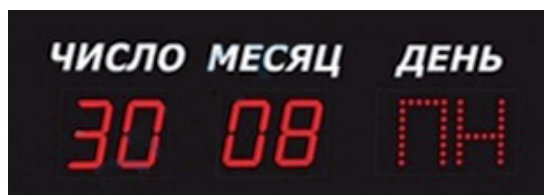


Рис. 9. Табло-календарь

Задача 2.

Написать программу, реализующую интерфейс электронного табло в банке или на почте, которое показывает номер очереди и номер окна, к которому надо подойти (рис. 10).



Рис. 10. Табло для очереди

(Рисунки 9 и 10 взяты с сайта: <https://www.distablo.ru/> и отредактированы.)

В статье рассмотрена реализация задачи создания текстового интерфейса с использованием языка программирования Python, хотя решение такой задачи можно осуществлять с использованием других языков программирования. Но в школе все чаще в качестве языка программирования изучается именно Python [8], в том числе из-за его удобства при решении задач ЕГЭ по информатике [11].

Важно включать изучение данного языка программирования в программу подготовки будущих учителей информатики, хотя нельзя отказываться и от языка Pascal [7], который изначально был предназначен для изучения алгоритмизации и программирования. При оценке предметных и методических компетенций выпускников педагогического вуза должны проверяться компетенции, связанные со знанием алгоритмизации и программирования [13].

Список источников

1. Босова Л. Л. О новых подходах к изучению школьной информатики в условиях цифровой трансформации общества // Информатика в школе. 2022. № 4. С. 5–14. EDN: DKRLZV. DOI: 10.32517/2221-1993-2022-21-4-5-14
2. Босова Л. Л., Аквилянов Н. А., Кочергин И. О., Штепа Ю. Л., Буцева Т. А. Информатика. 8–9 классы. Начала программирования на языке Python. Дополнительные главы к учебникам. М.: БИНОМ. Лаборатория знаний, 2022. 96 с.
3. Гаркавенко Г. В., Кубряков Е. А. Олимпиада по информатике в ВГПУ как средство популяризации предмета среди обучающихся // Актуальные проблемы прикладной математики, информатики и механики. Сборник трудов Международной научной конференции. Воронеж, 2019. С. 1393–1398. EDN: YYJWVF.

4. *Гаркавенко Г. В., Морозова В. В.* Пример исследовательской деятельности на уроках информатики // Актуальные проблемы методики обучения информатике и математике в современной школе. Материалы международной научно-практической интернет-конференции. М., 2021. С. 92–98. EDN: AJXOJE.
5. *Гаркавенко Г. В., Шеметов И. Д.* Некоторые особенности изучения языка программирования PYTHON // Информатика: проблемы, методы, технологии. Материалы XXIII Международной научно-практической конференции им. Э. К. Алгазиева. Воронеж, 2023. С. 1177–1183. EDN: TJIXFK.
6. *Добровольская Н. Ю., Харченко А. В.* Задания-исследования как способ развития навыков программирования // Информатика в школе. 2019. № 3. С. 48–51. EDN: SPUXPV. DOI: 10.32517/2221-1993-2019-18-3-48-51
7. *Зубрилин А. А.* Единый подход к подготовке будущих учителей информатики в ракурсе формирования цифровых компетенций // Информатика и образование. 2022. Т. 37. № 2. С. 42–49. EDN: VXCJRN. DOI: 10.32517/0234-0453-2022-37-2-42-49
8. *Каракозов С. Д., Маняхина В. Г.* Python как базовый язык обучения программированию в школе // Информатика в школе. 2020. № 1. С. 26–30. EDN: WDPXFF. DOI: 10.32517/2221-1993-2020-19-1-26-30
9. *Кольцова К. И.* Использование сюжетных задач при обучении программированию на Python // Информатика в школе. 2023. № 1. С. 7–12. EDN: GPEZPH. DOI: 10.32517/2221-1993-2023-22-1-7-12
10. *Кривоплясова Е. В., Нефёдова В. Ю., Прилепина А. В.* Методика обучения основам программирования на языке PYTHON // Информатика в школе. 2020. № 3. С. 24–30. EDN: NCKNDR. DOI: 10.32517/2221-1993-2020-19-3-24-30
11. *Маркелов В. К., Завьялова О. А.* Язык программирования Python как альтернативный инструмент для решения заданий ЕГЭ по информатике // Информатика в школе. 2023. № 2. С. 63–72. EDN: UBRWOS. DOI: 10.32517/2221-1993-2023-22-2-63-72
12. *Поляков К. Ю., Еремин Е. А.* Информатика. 9 класс. М.: БИНОМ. Лаборатория знаний, 2018. 288 с.
13. *Чудинский Р. М., Малев В. В., Малева А. А., Дубов В. М., Башарина С. О.* Оценка предметных и методических компетенций выпускников педагогического вуза — будущих учителей информатики: дефициты и перспективы // Информатика и образование. 2023. Т. 38. № 4. С. 44–56. EDN: MIKZRW. DOI: 10.32517/0234-0453-2023-38-4-44-56
14. Язык Python // Преподавание, наука и жизнь: сайт Константина Полякова. <https://kpolyakov.spb.ru/school/probook/python.htm>