

DOI: 10.32517/2221-1993-2024-23-5-12-18

А. Ю. Босова, А. И. Бутусова, А. О. Редькина

Московский педагогический государственный университет, г. Москва, Россия

МЕТОДИКА КОМПЛЕКСНОГО ИСПОЛЬЗОВАНИЯ ВИДЕОМАТЕРИАЛОВ И ОПОРНЫХ СХЕМ ПРИ ИЗУЧЕНИИ ПРОГРАММИРОВАНИЯ*

Аннотация

В статье обосновывается необходимость применения визуальных форм представления учебного материала с учетом особенностей современного поколения обучающихся. Приводятся описание и требования к созданию таких визуальных форм представления учебного материала, как видеоролики и опорные конспекты, отмечается их дидактический потенциал: сообщение и систематизация учебной информации; формирование наглядных представлений о явлениях, событиях и фактах; повышение информационной плотности занятий за счет ускоренной подачи информации; обеспечение эмоциональной насыщенности учебного материала и формирование на этой основе позитивного отношения обучающихся; индивидуализация процесса обучения. Дается описание комплекса визуальных средств обучения «Начала программирования на Python», включающего видеоролики и опорные конспекты. Приводится пример сценария урока комбинированного типа с комплексным использованием видеоролика и опорного конспекта.

Ключевые слова: учебный материал, визуализация, видеоролик, опорный конспект, информатика, программирование.

1. Введение

Стремительные темпы цифровизации многих сфер нашей жизни оказывают серьезное влияние на становление психических процессов обучающихся. Современное поколение молодых людей все чаще сталкивается с трудностями восприятия текстовой информации, отличается недостаточным уровнем развития способностей нахо-

дить и выделять существенные признаки, углубляться в информацию [6, 7]. Бурно развивающиеся социальные сети, стремящиеся к охвату и привлечению большого количества молодой аудитории в своем «быстро прокручиваемом» мире, вызывают эмоции и побуждают к действию преимущественно за счет визуальной коммуникации. Современным школьникам, выросшим в тесном контакте с компьютерными и медиатеchnологиями,

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_05-2024/

Контактная информация

Босова Анна Юрьевна, старший преподаватель кафедры математики и информатики в начальной школе, факультет начального образования, Институт детства, Московский педагогический государственный университет, г. Москва, Россия; *адрес:* 119991, Россия, г. Москва, ул. Малая Пироговская, д. 1, стр. 1; *e-mail:* abosova@gmail.com

Бутусова Анастасия Игоревна, студентка 5-го курса факультета начального образования, Институт детства, Московский педагогический государственный университет, г. Москва, Россия; *адрес:* 119991, Россия, г. Москва, ул. Малая Пироговская, д. 1, стр. 1; *e-mail:* ai_butusova@student.mpgu.edu

Редькина Александра Олеговна, студентка 5-го курса факультета начального образования, Институт детства, Московский педагогический государственный университет, г. Москва, Россия; *адрес:* 119991, Россия, г. Москва, ул. Малая Пироговская, д. 1, стр. 1; *e-mail:* ao_redkina@student.mpgu.edu

A. Yu. Bosova, A. I. Butusova, A. O. Redkina
Moscow Pedagogical State University, Moscow, Russia

METHODOLOGY FOR COMPREHENSIVE USE OF VIDEO MATERIALS AND REFERENCE SCHEMES IN THE STUDY OF PROGRAMMING

Abstract

The article substantiates the need to use visual forms of presenting educational material, taking into account the characteristics of the modern generation of students. The description and requirements for the creation of such visual forms of presenting educational material as videos and reference schemes are given, their didactic potential is noted: communication and systematization of educational information; formation of visual representations of phenomena, events and facts; increasing the information density of classes due to accelerated delivery of information; ensuring the emotional richness of educational material and the formation on this basis of a positive attitude of students; individualization of the learning process. A description of the complex of visual teaching aids "Beginnings of Python programming" is given, including videos and reference schemes. An example of a combined lesson scenario with the comprehensive use of a video and reference schemes is given.

Keywords: educational material, visualization, video, reference scheme, informatics, programming.

сотовой (мобильной) связью, присущи фрагментарное (клиповое) мышление и приоритетное восприятие визуальной информации; их привлекают, вызывая эмоциональный отклик и активизируя познавательные психические процессы, различные форматы динамического и статического визуального ряда.

В связи с изменениями, происходящими в мыслительных процессах обучающихся, одним из ключевых направлений цифровой трансформации образования становится разработка и активное использование на уроках различных визуальных форм представления учебного материала [9].

В данной статье мы остановимся на таких визуальных формах представления учебного материала, как видеоролик и опорный конспект. Они обладают значительными дидактическими возможностями, а именно:

- сообщение и систематизация учебной информации;
- формирование наглядных представлений о явлениях, событиях и фактах;
- повышение информационной плотности занятий за счет ускоренной подачи информации;
- обеспечение эмоциональной насыщенности учебного материала и формирование на этой основе позитивного отношения обучающихся;
- индивидуализация процесса обучения.

2. Видеоролики

Благодаря развивающимся технологическим решениям:

- сетевым технологиям, обеспечивающим высокую скорость доступа к видеоконтенту;
- доступному программному и аппаратному обеспечению;
- широким возможностям для воспроизведения;
- простоте интеграции видео в имеющиеся цифровые образовательные материалы

видеоматериалы находят все более широкое применение в учебном процессе [2].

Для использования в образовательном процессе наиболее эффективны видеоролики, время воспроизведения которых не превышает 10 минут; если видеозапись имеет большую длительность воспроизведения, необходимо предусмотреть способ навигации по ее содержательным частям (эпизодам). Эмоциональное вовлечение обучающихся может быть достигнуто за счет:

- использования дополнительных видеоматериалов, задающих ассоциативный зрительный ряд и динамичный темп изображения;
- сочетания привычных элементов со встроенными внешними;
- смены крупных и средних планов, отображающих богатую мимику лектора, обеспечивающих передачу эмоций и ценностей;
- концентрации внимания на образе лектора благодаря активизирующему внимание цветовому фону в помещении и отсутствию лишних деталей в кадре.

Использование видеоконтента оказывает безусловное влияние на трансформацию образовательного ландшафта, обеспечивая обучающимся возможность его просмотра не строго по расписанию и в классной комнате, а в любом

удобном для обучающихся месте, нужное количество раз и в любое подходящее им время. Перечисленные выше возможности видеоматериалов широко используются при проведении так называемых «перевернутых» уроков. В разделе 5 статьи мы покажем возможности интеграции видеоматериалов в традиционный очный урок.

3. Опорные конспекты

Понятие опорного конспекта и методика его использования были предложены педагогом-новатором В.Ф. Шаталовым в 80-х годах прошлого века [11]. **Опорный конспект** — это форма кодирования учебной информации в виде словесных, графических и картографических опор. Опорный конспект должен достаточно полно отражать содержание учебного материала и представлять его в определенной логической последовательности. При этом информация в опорном конспекте должна быть представлена лаконично и быть структурированной, т. е. разделенной на отдельные взаимосвязанные блоки, — для этого могут использоваться специально выделенные области, линии, стрелки и другие элементы. Акцентировать внимание на наиболее важных моментах в опорных конспектах рекомендуется с помощью шрифта, цвета, подчеркивания, графических образов. Желательно, чтобы опорный конспект был неординарен по графическому исполнению, благодаря чему он будет лучше сохраняться в памяти обучающихся. Опорные конспекты можно создавать с помощью графических редакторов и онлайн-ресурсов, исходя из навыков разработчика.

С помощью опорных конспектов обучающиеся могут наглядно представить учебный материал, увидеть логические связи между смысловыми блоками, научиться выделять главное и устанавливать причинно-следственные связи. В настоящее время наиболее близким к понятию опорного конспекта является понятие *инфографики* [1].

4. Использование видеороликов и опорных конспектов на уроках информатики

Особенно актуально применение таких средств обучения, как видеоматериалы и опорные конспекты, при изучении предметов, предполагающих освоение большого объема теоретической информации. Одним из таких учебных предметов является «Информатика».

Приоритетное внимание в процессе обучения информатике в школе уделяется тематике, связанной с алгоритмами и программированием. В рамках соответствующего тематического блока школьники рассматривают основные алгоритмические структуры, отрабатывают умения и навыки составления простых программ на одном из языков программирования высокого уровня [8, 10]. Для успешного освоения предполагаемых компетенций школьникам необходимо постоянно удерживать внимание одновременно на большом количестве используемых команд, при этом учитывать синтаксис языка и его особенности. Для большинства современных школьников такая задача становится непосильной.

Для облегчения процесса усвоения материала, начиная с изучения базовых основ программирования, мы

Комплекс визуальных средств обучения «Начала программирования на Python»

Обучающий видеоролик	Опорный конспект
Урок 1. Командный режим работы. Знакомство с Черепашкой	Приложение 1
Урок 2. Программный режим работы	Приложение 2
Урок 3. Создаем процедуры	Приложение 3
Урок 4. Круги, окружности и дуги	Приложение 4
Урок 5. Процедуры с параметрами	Приложение 5
Урок 6. Случайные числа и циклические алгоритмы	Приложение 6
Урок 7. Круги и циклы	Приложение 7
Урок 8. Первая игра	Приложение 8

предлагаем использовать представленные выше виды средств визуализации информации, а именно:

- 1) **комплект обучающих видеороликов «Начала программирования на Python»** по освоению учащимися VI—VII классов компьютерной графики на основе использования модуля turtle языка программирования Python, разработанный авторами УМК «Информатика» Л. Л. Босовой и А. Ю. Босовой [3–5]. В роли учителя в этих видеороликах выступает ученик седьмого класса Александр Босов, что создает дополнительное позитивное отношение к представляемому учебному материалу со стороны обучающихся. Все видеоролики размещены по адресу: <https://rutube.ru/plst/476792/>
- 2) **комплект опорных конспектов «Начала программирования на Python»**, разработанный студентами МПГУ А. И. Бутусовой и А. О. Редькиной; все опорные конспекты представлены в *Приложениях 1–8**.

* Приложения также можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_05-2024/

Полный состав комплекса визуальных средств обучения «Начала программирования на Python» представлен в таблице 1.

5. Пример урока с использованием видеоролика и опорного конспекта

В качестве примера рассмотрим первый урок «Командный режим работы. Знакомство с Черепашкой». Как и большинство уроков информатики, это урок комбинированного типа, на котором происходит изучение нового материала, предъявление новой информации, практическое применение полученных знаний и умений, а также их систематизация и обобщение. Такая многозадачность обеспечивается созданием подробного сценария урока, в котором предусмотрено деление видеоролика на отдельные учебные эпизоды (табл. 2).

В условиях работы в классе, когда предусматривается фронтальная работа с видеороликом, наличие опорного конспекта повышает уровень усвоения материала и обеспечивает возможность создания индивидуального образовательного маршрута обучающемуся.

Сценарий урока с использованием видеоролика и опорного конспекта

№ п/п	Этап урока	Тайм-код	Действия учителя	Действия учащихся
1	Мотивация	00:00	Демонстрирует ролик, показывает, где на рабочих компьютерах находится программа IDLE, предлагает ребятам открыть окно консоли.	Смотрят видео, выполняют открытие окна консоли IDLE.
2	Целеполагание	01:31	Демонстрирует ролик.	Смотрят ролик.
3	Решение поставленной задачи «Подключение библиотеки turtle»	01:57	Демонстрирует ролик, предлагает выполнить подключение библиотеки.	Смотрят ролик, выполняют подключение библиотеки.

Окончание табл. 2

№ п/п	Этап урока	Тайм-код	Действия учителя	Действия учащихся
4	Коррекция 1		При возникновении ошибок обращает внимание на написание команды, представленное в верхней части опорного конспекта.	Выполняют корректировку. Те, кто выполнил задание верно и не нуждается в корректировке, могут ознакомиться с другими командами, помещенными в опорный конспект.
5	Решение поставленной задачи «Ввод первой команды»	02:52	Демонстрирует ролик, предлагает ввести команду.	Смотрят ролик, вводят команду.
6	Целеполагание	03:18	Демонстрирует ролик, фокусирует внимание на написании команд выбора формы Черепашки в опорном конспекте.	Смотрят ролик, изучают варианты форм Черепашки.
7	Решение поставленной задачи «Изменение формы Черепашки»	03:31	Демонстрирует ролик, предлагает ввести команду по изменению формы Черепашки.	Смотрят ролик, выполняют команду, проводят эксперименты по изменению формы Черепашки.
8	Решение поставленной задачи «Изменение цвета черепашки»	04:13	Демонстрирует ролик, предлагает ввести команду по изменению цвета Черепашки, обращает внимание на написание команд выбора цвета Черепашки в опорном конспекте.	Смотрят ролик, выполняют команду, проводят эксперименты по изменению цвета Черепашки.
9	Решение поставленной задачи «Изменение размера Черепашки»	05:14	Демонстрирует ролик, предлагает ввести команду по изменению размера Черепашки, обращает внимание на написание команд установки размеров Черепашки в опорном конспекте.	Смотрят ролик, выполняют команду, проводят эксперименты по изменению размера Черепашки.
10	Решение поставленной задачи «Ввод команды для перемещения Черепашки»	05:54	Демонстрирует ролик, предлагает ввести команду по перемещению Черепашки, обращает внимание на написание команд для перемещения Черепашки в опорном конспекте.	Смотрят ролик, выполняют команду, проводят эксперименты по перемещению Черепашки по экрану.
11	Коррекция 2	06:34	Демонстрирует ролик, предлагает внести изменения в программу.	Смотрят ролик, вносят изменения.
12	Решение поставленной задачи «Очистка экрана»	06:49	Демонстрирует ролик, предлагает очистить экран.	Смотрят ролик, очищают экран.
13	Решение поставленной задачи «Рисуем прямоугольник»	07:09	Демонстрирует ролик, предлагает нарисовать прямоугольник с заданными размерами.	Смотрят ролик, рисуют прямоугольник с заданными размерами.
14	Решение поставленной задачи «Ввод команды stamp»	07:25	Демонстрирует ролик, предлагает выполнить команды по копированию Черепашки.	Смотрят ролик, выполняют копирование Черепашки.
15	Решение поставленной задачи «Ввод команды clearstamp»	08:25	Демонстрирует ролик, предлагает выполнить команды по удалению одной из копий Черепашки.	Смотрят ролик, выполняют удаление одной из копий Черепашки.
16	Подведение итогов	08:41	Демонстрирует ролик, проводит фронтальный опрос по названию команд, комментирует работы учеников.	Смотрят ролик, отвечают на вопросы, демонстрируют свои работы.
17	Объяснение домашнего задания	09:22	Демонстрирует ролик, поясняет домашнее задание.	Смотрят ролик, задают вопросы по домашнему заданию.
18	Рефлексия	09:29	Демонстрирует ролик, мотивирует на следующее занятие, получает обратную связь от учеников.	Смотрят ролик, делятся впечатлениями от урока.

Список источников

1. *Бондарева К. Д., Исаева М. А.* Инфографика в обучении информатике: эффективный способ представления материала для VII класса // Информатика в школе. 2024. Т. 23. № 1. С. 28–34. EDN: NIEHNJ. DOI: 10.32517/2221-1993-2024-23-1-28-34.
2. *Босова Л. Л.* Видео как современный формат представления образовательного контента // Информатизация образования и методика электронного обучения: цифровые технологии в образовании. Материалы V Международной научной конференции. В 2-х ч. (Красноярск, 21–24 сентября 2021 года). Ч. 2. Красноярск: СФУ, 2021. С. 437–441. EDN: CYMCAV.
3. *Босова Л. Л., Босова А. Ю.* Информатика. 6 класс. Базовый уровень: учебное пособие. М.: Просвещение, 2023. 240 с.
4. *Босова Л. Л., Босова А. Ю., Аквилянов Н. А.* Информатика. 7 класс. Углубленный уровень: учебник. В 2-х ч. Ч. 2. М.: Просвещение, 2024. 159 с.
5. *Босова Л. Л., Филиппов В. И., Босова А. Ю.* Программирование графики и анимации в курсе информатики основной школы // Информатика в школе. 2022. № 5. С. 5–15. EDN: LYUPWI. DOI: 10.32517/2221-1993-2022-21-5-5-15.
6. *Гаврилова И. В.* Учет клипового мышления обучающихся на примере изучения алгоритмических конструкций

в школьном курсе информатики // Личность и общество. 2019. № 2 (2). С. 17–19. EDN: LLLUSF. DOI: 10.32743/2658-6487.2019.2.2.69.

7. *Ломбина Т. Н., Юрченко О. В.* Особенности обучения детей с клиповым мышлением // Общество: социология, психология, педагогика. 2018. № 1. С. 92–95. EDN: YMPRII. DOI: 10.24158/spp.2018.1.7.

8. Приказ Министерства просвещения Российской Федерации от 18 мая 2023 г. № 370 (ред. от 19.03.2024) «Об утверждении федеральной образовательной программы основного общего образования». <https://docs.cntd.ru/document/1301798826>

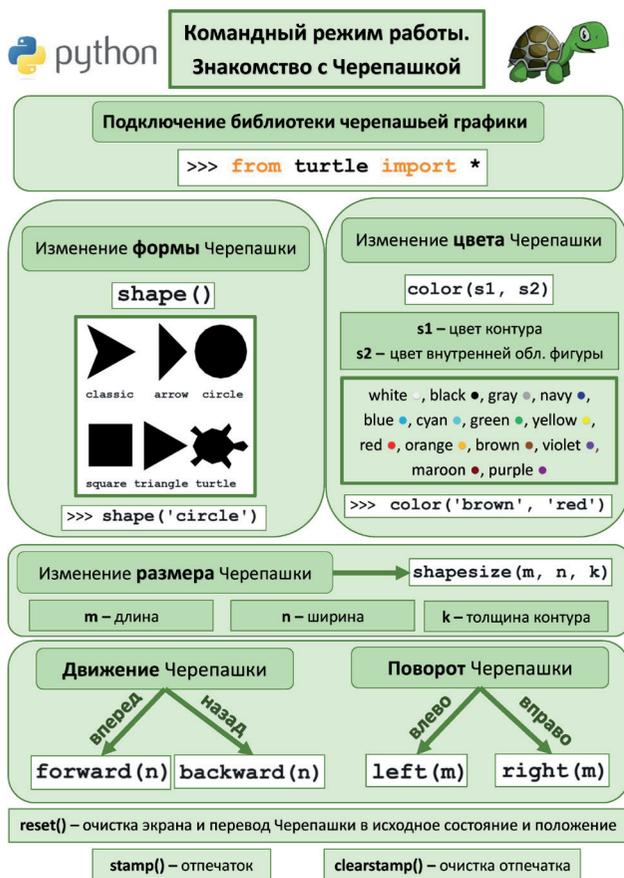
9. *Пузиновская С. Г., Счеснович О. А.* Лучше один раз увидеть, чем сто раз услышать, или Визуализация учебной информации на уроках информатики // Информатика в школе. 2022. № 5. С. 51–58. EDN: XENUZH. DOI: 10.32517/2221-1993-2022-21-5-51-58.

10. Федеральная рабочая программа основного общего образования. Информатика (базовый уровень) (для 7–9 классов общеобразовательных организаций). https://edsoo.ru/wp-content/uploads/2023/08/15_ФРП-Информатика-7-9-классы_база.pdf

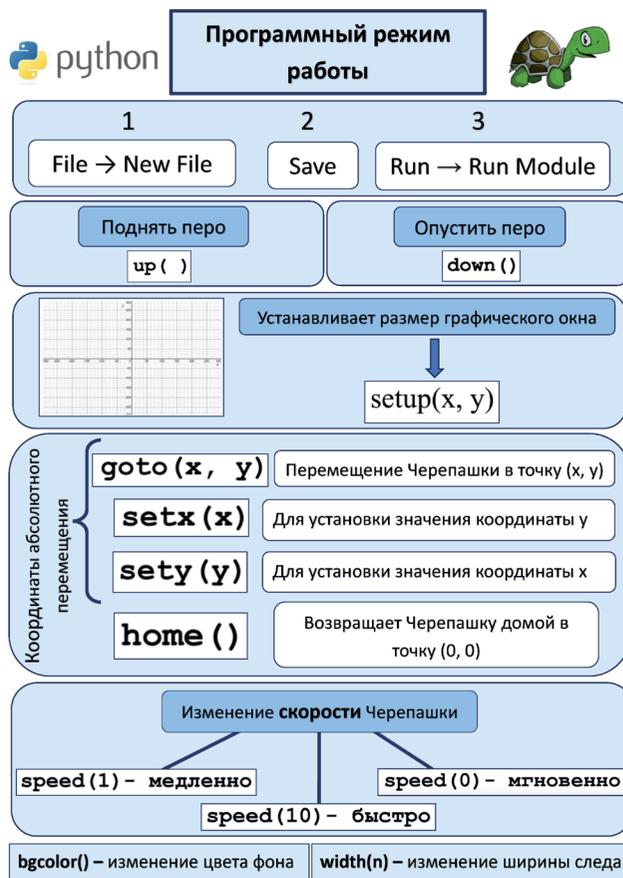
11. *Шаталов В. Ф.* Учить всех, учить каждого // Педагогический поиск / сост. И. Н. Баженова. М.: Педагогика, 1988. С. 141–204.

Приложения

Приложение 1



Приложение 2



Приложение 3

python Создаем процедуры

Исходное положение Черепашки

Вариант 1

```
from turtle import*
setup(600, 600)
fd(80)
lt(90)
fd(40)
lt(90)
fd(20)
fd(40)
rt(90)
fd(40)
lt(90)
fd(20)
```

Подходящий вариант, чтобы перенести рисунок в другую точку, используя up и down

Вариант 2

```
from turtle import*
setup(600, 600)
setx(80)
sety(60)
setx(60)
sety(20)
setx(0)
sety(0)
```

Неподходящий вариант для перемещения в другую точку

Команды абсолютного перемещения

```
goto(x, y)
setx(x)
sety(y)
```

Команды относительного перемещения

```
forward(n)
backward(n)
left(m)
right(m)
```

```
def angle():
    fd(80)
    lt(90)
    fd(60)
    lt(90)
    fd(20)
    lt(90)
    fd(40)
    rt(90)
    fd(60)
    lt(90)
    fd(20)
```

службное слово, название процедуры, двоеточие, отступ, тело процедуры

1) Все команды, образующие тело процедуры, пишутся с отступом в 4 пробела.

2) Основная программа записывается после процедуры. Вызывается процедура из основной программы по имени.

Приложение 4

python Круги, окружности и дуги

Для изображения кругов можно использовать команду: `dot(d, s)`

d – диаметр **s** – цвет

Изображение окружности

r – радиус

```
circle(r)
```

против часовой

```
circle(-r)
```

по часовой

Процедура, изображающая окружность, для которой известны координаты ее центра и радиус

```
def ring(x, y, r):
    up()
    setx(x)
    sety(y-r)
    down()
    circle(r)
```

Изображение дуг

```
circle(r, m)
```

r – радиус **m** – значение угла (в градусах)

Приложение 5

python Процедуры с параметрами

Процедуры для рисования прямоугольника и треугольника

```
def rectangle(x, y, a, b):
    up()
    goto(x, y)
    down()
    fd(a)
    lt(90)
    fd(b)
    lt(90)
    fd(a)
    lt(90)
    fd(b)
    lt(90)
```

```
def triangle(x, y, a):
    up()
    goto(x, y)
    down()
    fd(a)
    lt(120)
    fd(a)
    lt(120)
    fd(a)
    lt(120)
```

Пример основной программы

```
from turtle import*
setup(800, 800)
rectangle(-250, -150, 100, 350)
rectangle(-225, 125, 50, 50)
rectangle(-150, -150, 300, 250)
rectangle(-50, -150, 100, 100)
rectangle(-50, 100, 100, 100)
rectangle(150, -150, 100, 350)
rectangle(175, 125, 50, 50)
triangle(-250, 200, 100)
triangle(-50, 200, 100)
triangle(150, 200, 100)
```

Для закрасивания фигуры в модуле turtle используется команда:

```
begin_fill()
```

Для завершения закрасивания используется команда:

```
end_fill()
```

Изменение цвета заливки: `fillcolor("red")`

```
def rectangle(x, y, a, b, s):
    rectangle(175, 125, 50, 50, "blue")
```

s – параметр цвета

Приложение 6

python Случайные числа и циклические алгоритмы

Подключение модуля random

```
from random import *
```

```
x = randint(-300, 300)
y = randint(-300, 300)
d = randint(10, 60)
```

Команды, по которым компьютер задумывает в качестве значения x и y какие-нибудь целые числа

Циклический алгоритм

```
from turtle import *
setup(600, 600)
from random import *
for i in range(50):
    x=randint(-300, 300)
    y=randint(-300, 300)
    d=randint(10, 30)
    goto(x, y)
    dot(d, "red")
```

Оператор цикла с известным числом повторений

Он представляет собой следующую инструкцию:

- Начать отсчет с 0 и остановиться, не доходя до 50.
- Каждое отсчитанное значение сохранить в переменной i.
- Для каждого значения переменной i, от 0 до 49, всего 50 разных значений.

Модель цветопередачи

Для того чтобы красить пиксель на экране компьютера в тот или иной цвет, достаточно узнать значения яркости для базовых цветов.

`colormod(255)` – команда, позволяющая выбрать один из режимов цвета RGB



Круги и циклы



```
from turtle import *
setup(600, 600)
x = -200
y = 0
d = 40
up()
for i in range(8):
    goto(x, y)
    dot(d, "red")
    x = x + d
```

Центр круга

Рисуем красный круг с диаметром d

Задаем новые начальные значения

Задаем новое положение для центра следующего круга, увеличив текущее значение x на d

Изменение значения y

```
from turtle import *
setup(600, 600)
x=0
y=250
d=50
up()
for i in range(11):
    goto(x, y)
    dot(d, 'blue')
    y = y - 50
```

Изменение значений x и y

```
from turtle import *
setup(600, 600)
x=250
y=250
d=50
up()
for i in range(11):
    goto(x, y)
    dot(d, 'blue')
    x = x - 50
    y = y - 50
```

Концентрические круги

```
from turtle import *
from random import *
setup(600, 600)
x=0
y=0
d=50
up()
colormode(255)
for i in range(10):
    r = randint(0, 255)
    g = randint(0, 255)
    b = randint(0, 255)
    goto(x, y)
    dot(d, (r,g,b))
    d = d - 50
```



Первая игра



Объектно-ориентированное программирование

– это подход, при котором программа рассматривается как набор объектов, взаимодействующих друг с другом

Язык объектно-ориентированного программирования

КЛАСС: СОБАКА

Объект: пёс Юрико

```

class Dog:
    def bark(self):
        print("Гав")
    def eat(self):
        print("Еду")
    def sleep(self):
        print("Сплю")
    def color(self):
        print("Коричневый")
    def weight(self):
        print("10 кг")
    def breed(self):
        print("Лабрадор")
    def fur(self):
        print("Шерсть")
    def color(self):
        print("Коричневый")
    def weight(self):
        print("10 кг")
    def breed(self):
        print("Лабрадор")
    def fur(self):
        print("Шерсть")
    
```

Свойства (характеристики): порода, вес, размеры, шерстный покров, окрас и т. д.

Способы поведения: просить еду, бегать, лаять, спать и т. д.

Атрибуты

Методы

Библиотеки для создания игры

```
import turtle
```

– библиотека черепашьей графики

```
import random
```

– библиотека для генерации случайных чисел

Создание объекта (беговая дорожка)

```
ww = turtle.Screen()
ww = turtle.Screen()
ww.setup(1020, 100)
ww.bgcolor("green")
```

Создание объекта (черепашка)

```
one = turtle.Turtle("turtle")
```

```
one.hideturtle()
```

→ Спрятать черепашку

```
one.showturtle()
```

→ Показать черепашку

Исходное положение черепашек на экране

```

one = turtle.Turtle("turtle")
one.color("red")
one.hideturtle()
one.penup()
one.goto(-500, 20)
one.showturtle()

two = turtle.Turtle("turtle")
two.color("blue")
two.hideturtle()
two.penup()
two.goto(-500, -20)
two.showturtle()
    
```

Перемещение черепашек до правой границы беговой дорожки

Условие «пока»

```
while one.xcor() < 500 and two.xcor() < 500:
```

Связка «и»

```

    one.fd(random.randint(1, 10))
    two.fd(random.randint(1, 10))
    
```

Случайное перемещение вперед от 1 до 10 шагов

Условия выигрыша

```

if one.xcor() > two.xcor():
    print("Победили красные!")
if one.xcor() < two.xcor():
    print("Победили синие!")
if one.xcor() == two.xcor():
    print("Ничья!")
    
```