



Python. Модуль 2. Урок 1.

Методические указания

Тема: Введение в модуль tkinter. Особенности использования. Класс. Создание геометрических объектов. Заполнение цветом.

Цели урока:

- Обеспечить закрепление учащимися понятий: координаты, объект, метод.
- Обеспечить усвоение учениками понятий: класс, окно, канва, закрепление объекта.
- Отработать навыки:
 - набора текста и его форматирования;
 - работы с консольным приложением в среде разработки;
 - подключения модуля tkinter;
 - решения задач с координатной плоскостью;
 - решение задач с использованием Canvas;
 - решения задач с использованием методов `create_rectangle()`, `create_oval()`, `create_polygon()`, `pack()`.
- Создать условия для развития:
 - логического и алгоритмического мышления;
 - элементов творческой деятельности;
 - памяти;
 - критического мышления, культуры групповой самоорганизации;
 - коммуникативной культуры.
- Организовать самостоятельную работу учащихся, направленную на развитие алгоритмического мышления, познавательных интересов и навыков работы на компьютере.

Задачи урока:

- повторить и закрепить понятия, пройденные на предыдущих уроках;
- изучить подключение модуля tkinter;
- изучить работу метода `mainloop()`;
- изучить Canvas и создание графического окна;
- написать программу с использованием Canvas;
- изучить методы `create_rectangle()`, `create_oval()`, `create_polygon()`, `pack()`;
- отработать навык составления алгоритма с использованием `create_rectangle()`, `create_oval()`, `create_polygon()`, `pack()`;
- написать программу с использованием `create_rectangle()`, `create_oval()`, `create_polygon()`, `pack()`.



Перед началом урока:

1. Открыть страницу с логинами и паролями учеников.
2. Открыть и запустить на проекторе презентацию (на слайдах присутствует gif анимация. Для её отображения запустить презентацию).
3. Запустить тестовую программу.
4. Проверить наличие раздатки.

План урока:

- [1. Анонс занятия \(2 мин.\).](#)
- [2. Повторение \(5 мин.\).](#)
- [3. Знакомство с модулем tkinter \(5 мин.\).](#)
- [4. Объекты. Класс. Первая программа \(10 мин.\).](#)
- [5. Элементы интерфейса \(15 мин.\).](#)
- [6. Координаты на холсте \(15 мин.\).](#)
- [7. Графические примитивы в tkinter \(10 мин.\).](#)
- [8. Выполнение самостоятельного задания на компьютере \(15 мин.\).](#)
- [9. Заливка фигур \(10 мин.\).](#)
- [10. Рефлексия \(3 мин.\).](#)
- [11. Задание на дом \(5 мин.\).](#)



1. Анонс занятия (2 мин.).

Слайд 2. Соберите детей в свободной от компьютеров зоне и обсудите, с какими понятиями им сегодня предстоит работать. С какими понятиями они уже знакомы, а что станет для них новым.

2. Повторение (5 мин.).

Слайд 3. Вспомните с учениками, что они изучали на прошлых занятиях, с какими понятиями познакомились.

3. Знакомство с модулем tkinter (5 мин.).

Слайд 4. Познакомьте детей с новым модулем. Обратите внимание на его аббревиатуру. Модуль tkinter позволит нам создавать графический интерфейс приложений. Подключение модуля точно такое же, как и у turtle.

Слайды 5-6. Предложите ученикам вспомнить любое приложение, которое они используют в телефоне или компьютере, какие элементы в них всегда присутствуют: кнопки, поля для ввода, картинки и т.п.

Слайды 7-8. Где же все это отображается? Когда мы запускаем приложение, что появляется? Появляется окно приложения. Это важный элемент любого интерфейса, его основа.

4. Объекты. Класс. Первая программа (10 мин.).

Слайды 9-10. Вспомните с детьми, как создаются объекты в Python. Задаётся имя объекта и через знак равно его принадлежность. Так как мы перешли к новому модулю, то окно приложения (root - корень) будет объектом Tk().

Слайд 11. Добавляем понятие класса. Когда мы говорим о принадлежности объекта, мы относим объект к какому-то классу. Класс определяет тип объекта и набор его методов. Класс Tk() является базовым и создает окно приложения, соответственно, через объект этого класса мы можем управлять нашим окном.

Слайд 12. Разберите с детьми первую программу. Что, по их мнению, должно появиться? Отправьте детей за компьютеры и попросите написать первую программу с использованием модуля tkinter. Соотнесите предположения и реальность. Что напоминает эта ситуация? Точно так же не появлялось окно с черепашкой на первом занятии. Как мы решили эту проблему?



Слайд 13. В отличие от turtle мы будем использовать метод `mainloop()`. Фактически метод не даёт разрушиться созданному объекту. То есть окно создаётся и остаётся, не пропадая. При этом написанная программа не останавливается, а продолжает работать и реагировать на действия пользователя. Например, на нажатие клавиш, мыши и т.п.

5. Элементы интерфейса (15 мин.).

Слайды 14-15. Разберите с детьми состав окна приложения. Само окно - это база, на которую можно что-то закреплять. Так как это графический интерфейс, то нам нужен холст, как настоящим художникам, только рисовать мы будем кнопками и другими элементами. Обратите внимание детей на то, что холст прикреплен кнопками к доске и зачем это сделано? Чтобы холст не упал. Это важный момент в обсуждении. В модуле `tkinter` холст - это класс `Canvas`. На нём можно располагать элементы приложения. При создании объекта класса `Canvas` мы должны указать: к какому окну относится этот холст, так как может быть многооконное приложение и программе надо понимать, чей это холст; размеры холста - его ширина и длина. Они указываются через знак `=` для соответствующих слов.

Слайд 16. Попросите детей добавить в свою программу создание холста и запустить программу.

Слайд 17-18. Почему мы не увидели холст? Ещё раз обратите внимание на кнопки. Мы создали холст, но не закрепили его, фактически он упал с нашей доски.

Слайд 19. Для закрепления объекта используется метод `pack()` - сборщик объекта. Для понятийности используйте закрепление элемента на холсте.

Слайд 20-21. Измените программу и проверьте, что появилось окно с холстом размером 640*480. Поинтересуйтесь у детей, могли ли они встречаться ранее с таким размером окна.

6. Координаты на холсте (15 мин.).

Слайды 22-23. Вспомните с детьми, откуда начинался отсчёт координат в окне черепашки.

Слайд 24. В отличие от модуля `turtle`, в модуле `tkinter` объекты создаются сразу. Достаточно указать место создания для объекта через координаты. Первое, что мы научимся делать - это рисовать линию. В команде указываем 4 числа: первые два - координаты начала линии, вторые два - координаты конца линии. То есть рисуем отрезок. Линия появляется сразу, рисование линий теперь мы видеть не будем.



Слайд 25. Предложите детям нарисовать на листе линию по указанным координатам. Точка начала для них сейчас находится в центре листа, 100-100 - верхний правый угол.

Слайды 26-27. После выполнения задания на листе, добавьте вызов этого же метода в программу и сравните результаты. Почему направление линий не совпали?

Слайды 28-29. Откуда берёт начало линия? В окне tkinter начало координат располагается в верхнем левом углу.

Слайд 30. Разберите направление осей в окне tkinter. Ось X направлена только вправо, ось Y только вниз. То есть в этом окне нет отрицательных координат.

7. Графические примитивы в tkinter (10 мин.).

Слайды 31-33. Для того, чтобы нарисовать прямоугольник достаточно указать координаты его расположения. Верхний левый угол и нижний правый, по этим координатах программа сама построит линии и отобразит такой прямоугольник на экране. Добавьте в программу метод для создания прямоугольника в указанных координатах и проверьте её работу.

Слайды 34-36. Круг - единственная фигура, которую мы не могли нарисовать черепашкой, потому что всегда ходили линиями. Подробно разберите, как программа рисует круг. Метод `create_oval` рисует в указанном прямоугольнике круг или овал. Если через координаты строится квадрат - значит программа нарисует круг, если был прямоугольник, то будет овал. Выполните задание и проверьте результат.

8. Выполнение самостоятельного задания на компьютере (15 мин.).

Слайд 37. Раздайте детям лист М2У1 раздатка 1. На листе указаны координаты и линии до объектов. Некоторые координаты дети должны рассчитать самостоятельно, обсудите расстояние между делениями.

Слайд 38. Проверьте программу вместе с детьми. Порядок создания объектов может отличаться.

Слайд 39-40. После выполнения задания с домом, предложите детям нарисовать машину в указанных координатах.

**Дети должны загрузить файл с программой к заданию
через платформу**



9. Заливка фигур (10 мин.).

Слайд 41. Разберите с детьми настройку fill. С её помощью мы можем заполнить фигуру цветом. Но крыша у нас сделана линиями - это не конечная фигура. Поэтому её заполнить цветом будет невозможно.

Слайд 42. Разберите с детьми метод create_polygon - создаёт многоугольник, достаточно указать координаты всех углов. Крыша имеет форму треугольника - 3 пары координат.

Слайд 43. Раздайте детям таблицу цветов. Выдайте задание с раскраской фигур. Обратите внимание на порядок создания фигур. Если сначала создан круг, а потом крыша, то заливка крыши перекроет заливку круга и он просто пропадёт для нас. Для того, чтобы это не происходило, достаточно поменять местами строки кода.

Слайд 44. Цвета, которые будут использовать дети могут отличаться от слайда, так как имеется большое множество оттенков, главное соблюсти цветовую гамму.

10. Рефлексия (3 мин.).

Слайд 45. Повторите материал занятия. Не забудьте загрузить проекты детей на склад.

«Отлично! Вы все большие молодцы! Но не забывайте, что только при регулярной практике и работе дома вы сможете добиться поставленных целей. Всем спасибо, жду вас на следующем занятии!»

11. Выдача задания на дом (5 мин.)

Слайды 46-47. Задание с двумя уровнями сложности:

1. Базовое, определяет степень усвоения нового материала.
2. Расширенное, связанное с материалами прошлых занятий.