



Методические рекомендации по теме

«Рисование символами: строка символов, основные операции и функции»

Цель:

- понимание смысла псевдографики и создание символами арт объектов на языке Python.

Задачи:

- расширение кругозора обучающихся в области информатики и программирования;
- знакомство с понятием «псевдографика», использование символов для создания арт объектов;
- знакомство с основами создания символьной псевдографики на языке Python;
- решение программных прикладных задач при создании псевдографики на языке Python;
- ранняя профориентация школьников, профессиональная деятельность программиста;
- развитие интеллектуальных способностей, логического и критического мышления.

Планируемые результаты

Личностные: обучающиеся получают навыки активной коммуникации в группе, осознанной ориентировки в мире ИТ профессий, постановки собственных образовательных задач и владение первичными навыками деятельностного анализа и критической оценки получаемой информации.

Предметные: обучающиеся получают представления: о создании псевдографики в языке Python; об основных операциях при рисовании символами на языке программирования «Python»; о прикладном использовании изученных операций и инструментов для рисования символами в программных проектах; о возможностях и особенностях применения рисования символами в практике работы программиста.

Метапредметные: обучающиеся получают возможность владения обще предметными понятием «символ», «псевдографика»; владение информационно-логическими умениями; владение умениями самостоятельно планировать пути достижения целей; умениями принятия решений и осуществления осознанного выбора; повысят уровень ИКТ-компетентности.

Материалы к занятию

Приложение 1: Сценарный план видеоролика

Приложение 2: Домашние задание и практика

Приложение 3: Краткие организационно-методические рекомендации по организации работы на занятии

Ход проведения урока

1. Организационный момент.

Мотивация на учебную деятельность.

Приветствие учащихся, сообщение темы и целей занятия (мы узнаем, об интересном виде искусства «псевдографике» – рисовании и создании анимации с помощью символов, научимся автоматизировать процесс создания «шедевров» в технике «псевдографики»).

Проблемная дискуссия по вопросам:

- Почему мог появиться такой вид искусства как «псевдографика»?
- Какие изображения проще всего нарисовать с помощью псевдографики? Почему?
- Может ли использование циклов упростить создание компьютерной псевдографики?

Итоги дискуссии (обобщаются преподавателем и фиксируются ответы учеников на доске, чтобы вернуться к ним и оценить правильность предположений учеников на этапе рефлексии):

- псевдографика отдельный вид искусства, который получил новое развитие с появлением программирования;
- возможности программирования позволяют нам создавать произведения в технике псевдографики проще и масштабнее.

Преподаватель называет ученикам тему и цели урока.

2. Вводный блок.

Тема.

Преподаватель при необходимости останавливая трансляцию, комментируя дополнительно тему занятия.

**см. сцены 1 – 2 (здесь и далее приводится Таблица «Содержание видеоролика». Приложение 1).*

3. Блок повторения.

Блиц-опрос.

Преподаватель предлагает ученикам ответить на **5 вопросов** по предыдущей теме; задания выполняются в сопровождении видеоролика с использованием таймера; ученики выполняют задания, голосуют, обсуждают результаты. Процедура голосования определяется инструкцией в сцене 3; учитель должен убедиться, что всем понятна процедура голосования. *Преподаватель может поставить ролик на паузу и обсудить результаты голосования; объяснить правильный ответ руководствуясь материалами предыдущего занятия*

**см. сцены 3 – 7*

4. Теоретический блок.

Псевдографика.

Продолжение демонстрации ролика с дальнейшим обсуждением вопросов:

- Какие особенности создания псевдографических произведений с точки зрения программирования?
- Для чего в учебном проекте используется переменная «Т»?

При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу; если ответы на вопросы вызывают у учеников затруднения, преподаватель может вывести нужную сцену ролика на экран для помощи ученикам.

**см. сцена 8 – 9*

5. Блок заданий.

Проект: «Сердце».

К началу демонстрации блока заданий ученики должны занять рабочие места и запустить Python (терминал IDLE) на своих компьютерах.

«Сердце»: включает практическое задание на написание программы для рисования символами.

После выполнения задания ученики получают работающий программный продукт – изображение сердца в технике псевдографики.

На сцене разбора задания преподаватель ставит ролик на паузу и вместе с учениками проводит разбор задания.

**см. сцены 10 – 17*

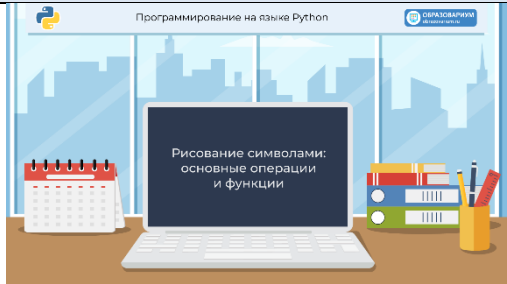

6. Рефлексия. Сообщение домашнего задания.

Завершаем демонстрацией ролика и кратким обобщением материалов занятия. Преподаватель возвращается к зафиксированному в ходе дискуссии в начале урока предположениям учеников и обсуждает насколько их предположения были правильными, делаются выводы.

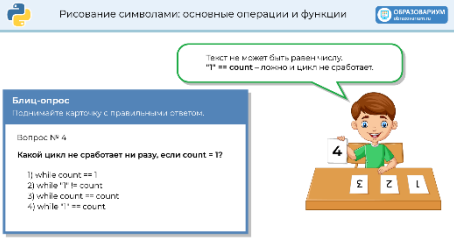
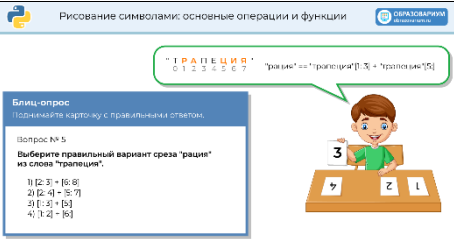
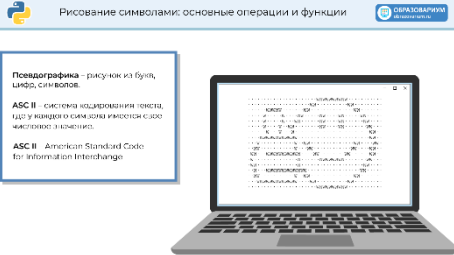
Преподаватель дает ученикам домашнее задание к следующему занятию (*Приложение 2*). **см. сцена 18*

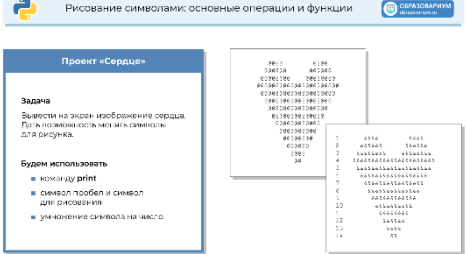
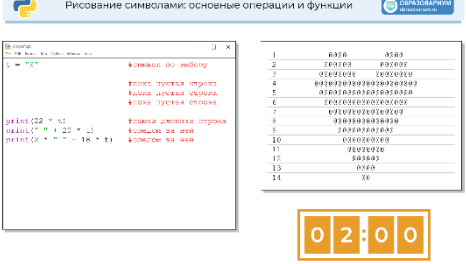
В таблице «Содержание видеоролика» представлен краткий текст из видеоролика, примеры заданий и задач, которые будут демонстрироваться на экране. Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры заданий и задач для использования в работе с учениками. Распечатанные тексты и задания из таблицы также можно применять в качестве раздаточного материала как на уроке, так и для домашних заданий.

Таблица. Содержание видеоролика

Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
Вводный блок. Мы узнаем	<p><i>Обозначаем ученикам тему и цели урока.</i></p> <p>Рисование символами: строка символов, основные операции и функции</p>	 <p>Сцена 1</p>	1 2
	<p>Искусство рисование символами появилось еще в далеком 19 веке благодаря появлению первых пишущих машинок, в 20 веке появился даже специальный термин – Typewrite art. И когда в пишущих машинках появились ленты разного цвета – стало возможным создавать даже цветные картинки.</p> <p>Мы тоже сегодня мы попробуем создать свою первую программу, используя псевдографику</p>	 <p>Сцена 2</p>	

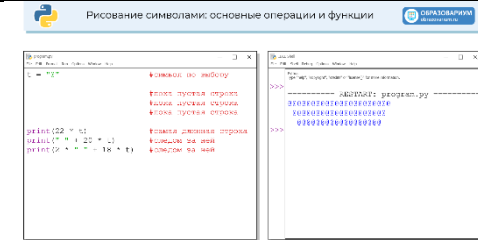
<p>Блок повторения.</p> <p>Блиц-опрос</p>	<p><i>Повторение материала предыдущего урока; на столе имеются пронумерованные карточки; после каждого вопроса выбираем ту, номер которой, совпадает с правильным ответом.</i></p> <p>Первый вопрос. Какое последнее значение будет у счетчика? Поднимите карточку с номером ответа.</p> <p>for i in range (1,4,4)</p> <p><i>Ответ 1. Шаг больше, чем интервал между старт и стоп. Цикл сработает только один раз со значением счетчика, равным старт.</i></p>	<p>Рисование символами: основные операции и функции</p> <p>Шаг больше, чем интервал между старт и стоп. Цикл сработает только один раз со значением счетчика, равным старт.</p> <p>Блиц-опрос Поднимайте карточку с правильным ответом.</p> <p>Вопрос №1 Какое последнее значение будет у счетчика? for i in range (1, 4, 4)</p> <p>Поднимите карточку с соответствующим числом.</p> <p>Сцена 3</p>	<p>3</p> <p>4</p> <p>5</p> <p>6</p> <p>7</p>
	<p>Второй вопрос. Какая функция может выводить не целое число?</p> <p>1) int 2) len 3) floor 4) round</p> <p><i>Ответ 4. round() может выводить дробное число с указанной точностью.</i></p>	<p>Рисование символами: основные операции и функции</p> <p>round() может выводить дробное число с указанной точностью.</p> <p>Блиц-опрос Поднимайте карточку с правильным ответом.</p> <p>Вопрос №2 Какая функция может выводить не целое число?</p> <p>1) int 2) len 3) floor 4) round</p> <p>Сцена 4</p>	
	<p>Третий вопрос. Какая строка содержит правильную запись увеличения num на count.</p> <p>1) num + count == num 2) num == num + count 3) num = num + count 4) num + count = num</p> <p><i>Ответ 3. Знак присвоения обозначается так « = ». Переменной присваиваем результат сложения старого значения и другой переменной: num = num + count.</i></p>	<p>Рисование символами: основные операции и функции</p> <p>Знак присвоения обозначается так « = ». Переменной присваиваем результат сложения старого значения и другой переменной: num = num + count.</p> <p>Блиц-опрос Поднимайте карточку с правильным ответом.</p> <p>Вопрос №3 Какая строка содержит правильную запись увеличения num на count?</p> <p>1) num + count == num 2) num == num + count 3) num = num + count 4) num + count = num</p> <p>Сцена 5</p>	

	<p>Четвертый вопрос. Какой цикл не сработает ни разу, если <code>count = 1</code>?</p> <ol style="list-style-type: none"> 1) <code>while count == 1</code> 2) <code>while "1" != count</code> 3) <code>while count == count</code> 4) <code>while "1" == count</code> <p><i>Ответ 4. Текст не может быть равен числу. "1" == count – ложно и цикл не работает.</i></p>	 <p>Рисование символами: основные операции и функции</p> <p>Блиц-опрос: Поднимайте карту(ы) с правильным(ыми) ответом(ами).</p> <p>Вопрос № 4 Какой цикл не сработает ни разу, если <code>count = 1</code>?</p> <ol style="list-style-type: none"> 1) <code>while count == 1</code> 2) <code>while "1" != count</code> 3) <code>while count == count</code> 4) <code>while "1" == count</code> <p>Текст не может быть равен числу. "1" == count – ложно и цикл не работает.</p> <p>Сцена 6</p>	
	<p>Пятый вопрос. Выберите правильный вариант среза "рация" из слова "трапеция"</p> <ol style="list-style-type: none"> 1) <code>[2:3] + [6:8]</code> 2) <code>[2:4] + [5:7]</code> 3) <code>[1:3] + [5:]</code> 4) <code>[1:2] + [6:]</code> <p><i>Ответ 3. "рация" == "трапеция"[1:3] + "трапеция"[5:]</i></p>	 <p>Рисование символами: основные операции и функции</p> <p>Блиц-опрос: Поднимайте карту(ы) с правильным(ыми) ответом(ами).</p> <p>Вопрос № 5 Выберите правильный вариант среза "рация" из слова "трапеция".</p> <ol style="list-style-type: none"> 1) <code>[2:3] + [6:8]</code> 2) <code>[2:4] + [5:7]</code> 3) <code>[1:3] + [5:]</code> 4) <code>[1:2] + [6:]</code> <p>"ТРАПЕЦИЯ" "рация" == "трапеция"[1:3] + "трапеция"[5:]</p> <p>Сцена 7</p>	
<p>Теоретический блок.</p> <p>Псевдо-графика.</p>	<p><i>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</i></p> <p>В интернете часто встречаются рисунки, сделанные при помощи букв, цифр, символов. Это так называемая псевдографика.</p> <p>Она называется ASC II (система кодирования текста, где у каждого символа имеется свое числовое значение).</p> <p>Полное его название звучит как American Standard Code for Information Interchange</p>	 <p>Рисование символами: основные операции и функции</p> <p>Псевдографика – рисунок из букв, цифр, символов.</p> <p>ASC II – система кодирования текста, где у каждого символа имеется свое числовое значение.</p> <p>ASC II: American Standard Code for Information Interchange</p> <p>Сцена 8</p>	<p>8 9</p>

	<p>Рассмотрим проект (относительно несложный) – это сердце, составленное из одинаковых символов. Заложим в программу возможность эти символы менять по нашему желанию.</p> <p>Разберем из чего состоит этот рисунок: это 14 строк, напечатанных при помощи команды принт, почти в каждой строке имеется определенное количество символов ПРОБЕЛ и символов, которые являются основой для рисования.</p> <p><i>Чтобы не писать многократно повторяющиеся символы удобнее использовать умножение символа на число.</i></p> <p>Начнем с создания текстовой переменной, которую будем использовать вместо символа. Нужно это для того, чтобы можно было легко поменять изображение не переписывая код.</p>	 <p>Сцена 9</p>	
<p>Блок заданий.</p> <p>Практические задания:</p> <p>Задание 1</p>	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p>Задание 1. Алгоритм «Сердце»</p> <p>Начнем с создания текстовой переменной, которую будем использовать вместо символа. Нужно это для того, чтобы можно было легко поменять изображение, не переписывая код.</p> <p>Назовем ее литерой Т и присвоим какой ни будь символ.</p> <p>После этого пропустим три строчки и начнем с самой длинной строки рисунка, номер четыре. Так будет удобнее считать. В ней 22 символа.</p> <p>Выведем их на печать.</p> <p>Перейдем на строчку вниз. Как несложно заметить, там сначала идет ПРОБЕЛ, а потом 20 символов.</p> <p><i>После запуска таймера учащиеся записывают код программы и проверяют её работу.</i></p>	 <p>Сцена 10</p>	<p>10</p> <p>11</p> <p>12</p> <p>13</p> <p>14</p> <p>15</p> <p>16</p> <p>17</p>

Разбор задания 1.

Убедившись, что программа выводит символы в заданной форме рисунка, продолжаем работать над созданием полного изображения.

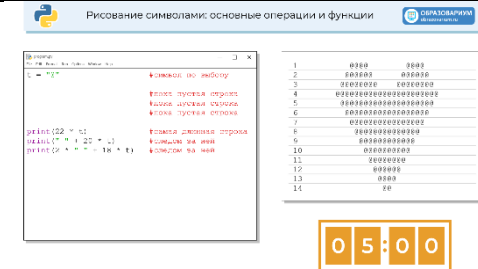


Сцена 11

После окончания дикторского текста запускается таймер на 5 мин.

Задание 1. Алгоритм «Сердце» (продолжение)

В продолжении кода программы допишите оставшиеся ниже семь строчек самостоятельно.

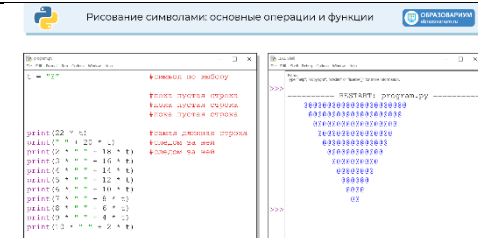


Сцена 12

Разбор задания 1. (продолжение)

Ваш код выглядит следующим образом:

```
print(3 * " " + 16 * t)
print(4 * " " + 14 * t)
print(5 * " " + 12 * t)
print(6 * " " + 10 * t)
print(7 * " " + 8 * t)
print(8 * " " + 6 * t)
print(9 * " " + 4 * t)
print(10 * " " + 2 * t)
```



Сцена 13

После окончания дикторского текста запускается таймер на 3 мин.

Задание 1. Алгоритм «Сердце» (окончание)

Добавим весь код, кроме первой строки, в специально созданный цикл с условием.

Полный алгоритм

Создать переменную t

Цикл с условием t не равно ""

Рисование узора

Запросить новый символ и

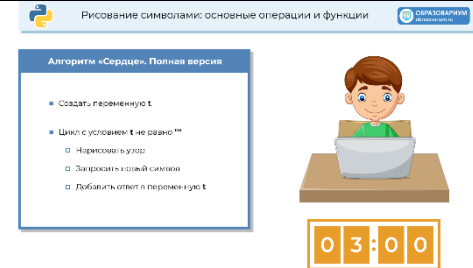
Добавить ответ в переменную t

Выходом из нее станет пустой ввод.

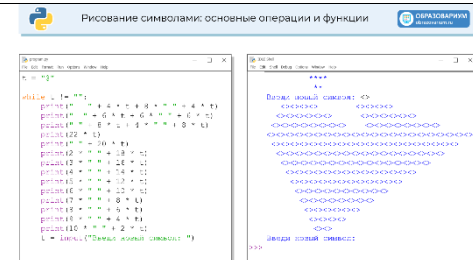
Разбор задания 1 (окончание). Полный код программы может выглядеть так:

```
t = "@"
while t != "":
    print(" " + 4 * t + 8 * " " + 4 * t)
    print(" " + 6 * t + 6 * " " + 6 * t)
    print(" " + 8 * t + 4 * " " + 8 * t)
    print(22 * t)
    print(" " + 20 * t)
    print(2 * " " + 18 * t)
    print(3 * " " + 16 * t)
    print(4 * " " + 14 * t)
    print(5 * " " + 12 * t)
    print(6 * " " + 10 * t)
    print(7 * " " + 8 * t)
    print(8 * " " + 6 * t)
    print(9 * " " + 4 * t)
    print(10 * " " + 2 * t)
    t = input("Введи новый символ ")
```


Теперь мы можем экспериментировать с рисунком вводя различные символы не прерывая программу.



Сцена 16



Сцена 17

<p>Блок завершения занятия. Рефлексия. Сообщение домашнего задания</p>	<p><i>Завершаем демонстрацией ролика и кратким обобщением материалов занятия.</i></p> <p>Подведем итоги.</p> <p>Мы узнали:</p> <ul style="list-style-type: none"> ■ что существует искусство рисования символами; ■ что в программировании псевдографика – это рисунок из букв, цифр, символов <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 18</p>	<p>18</p>
--	--	---	-----------

Приложение 2

Домашнее задание

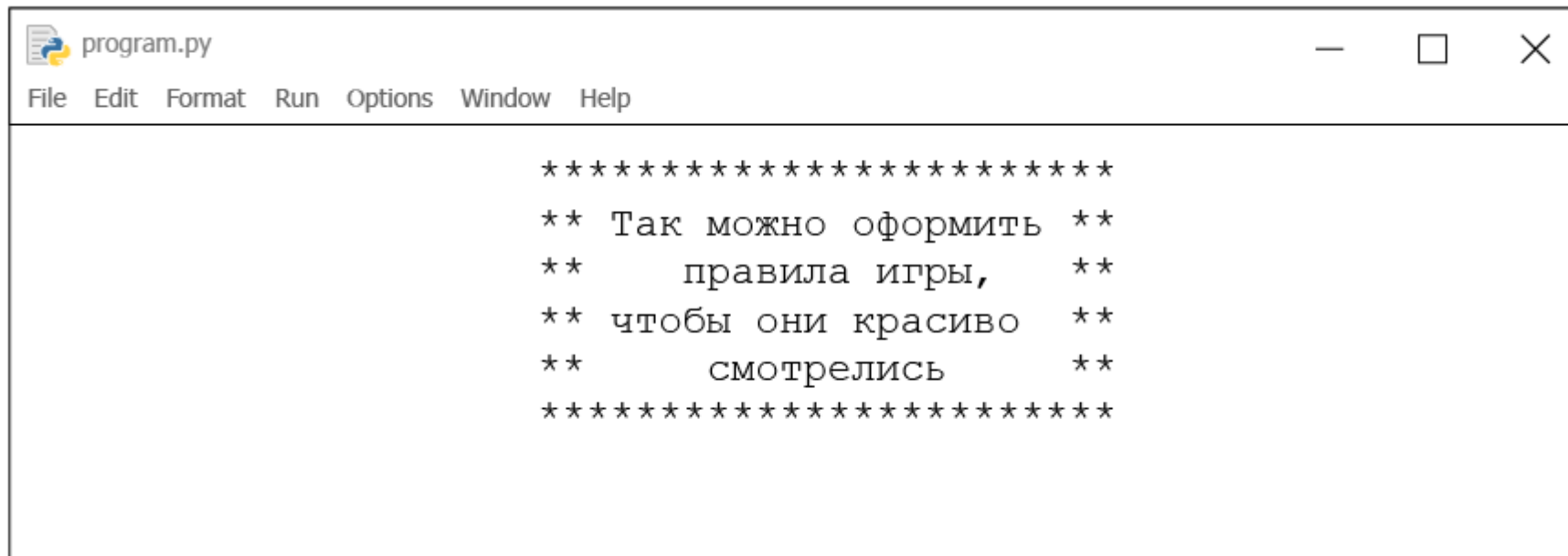
Поищите в интернете несложные картинки в технике «псевдографика», которые можно было бы реализовать с помощью программирования.

Практика

Проект «Заставка»

С помощью рисования символами можно красиво оформить правила работы с программой. Это может понадобиться, например для оформления правил игры в будущих проектах.

Потренируйтесь в оформлении правил, запрограммируйте вывод на экран такого текста.

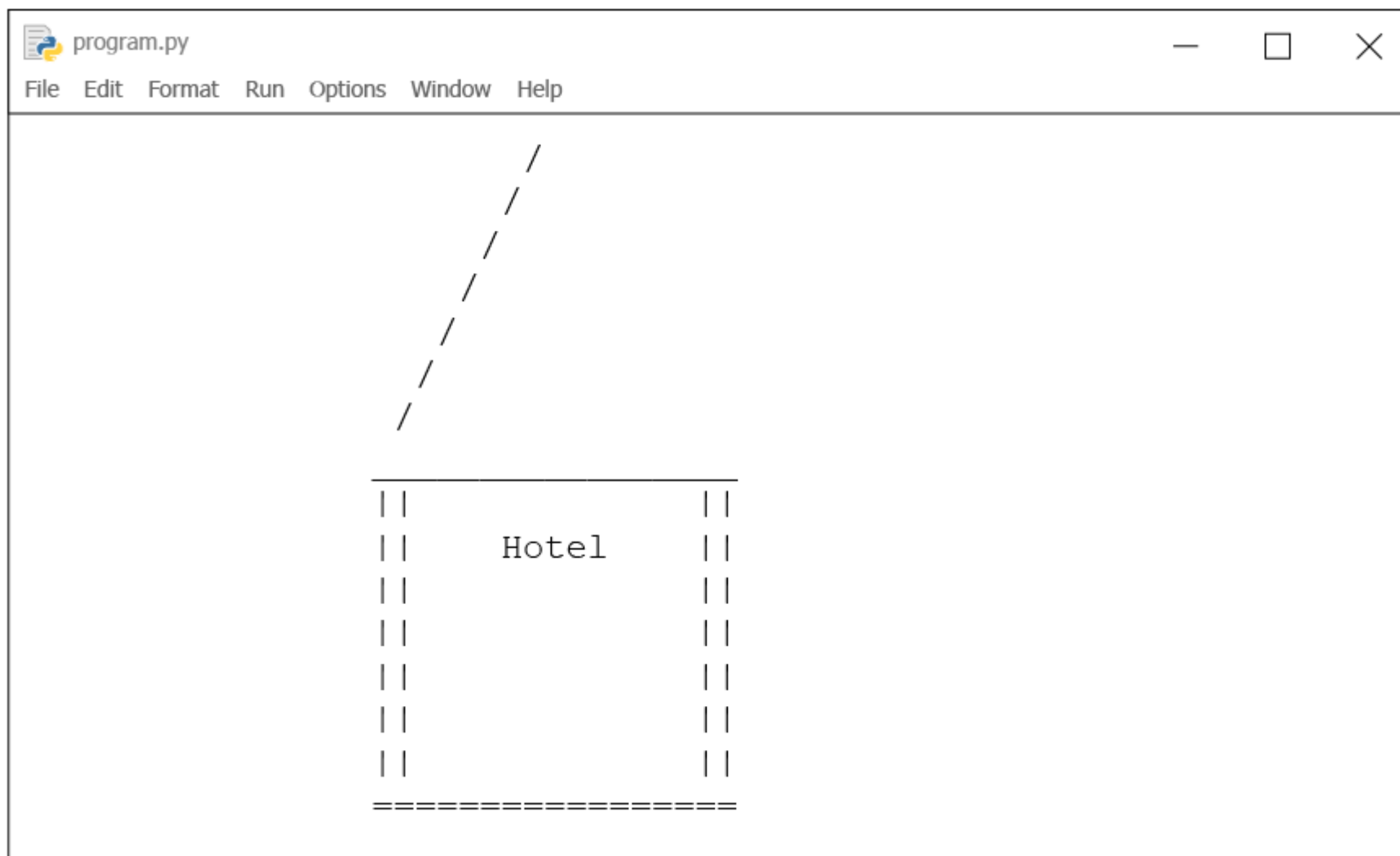


```
program.py
File Edit Format Run Options Window Help

*****
** Так можно оформить **
**   правила игры,   **
** чтобы они красиво **
**   смотрелись     **
*****
```

Проект «Домик»

Запрограммируйте вывод на экран такого изображения.



По желанию доделайте крышу, печную трубу, окошко... Можно увеличить размеры домика.

Имейте ввиду, символ « \ » использовать нельзя.

Краткие организационно-методические рекомендации по организации работы на занятии

«Рисование символами: строка символов, основные операции и функции».

В начале занятия можно рассмотреть домашнее задание – алгоритм бесконечного вклада. В процессе разбора можно сделать акцент на том, что существуют простые и сложные проценты. И как можно переделать алгоритм под ежемесячные начисления, при помощи бесконечного и конечного циклов.

Перед просмотром блока повторения из ролика необходимо раздать дидактический материал для выполнения заданий из блока повторение (по 4 пронумерованных карточки)

Во время голосований карточками можно останавливать ролик и вести учет правильных ответов. По окончании блока – отметить тех, у кого наилучший результат.

Далее карточки необходимо собрать.

После просмотра теоретического материала можно продемонстрировать ребятам примеры различных проектов в стиле псевдографика. Также можно запустить какой-нибудь генератор рисунков, чтобы сравнить получившийся результат с исходным материалом.

В момент разбора рисунка необходимо убедиться, что ребятам хорошо видна его структура – символы и пробелы.

Начинать программирование легче с самой длинной строки, которая выступает как образец. Это единственная строка, где отсутствует символ пробел! И обязательно напомните о возможности умножать текстовую информацию на целое число.

Если останется время – можно попробовать нарисовать простую геометрическую фигуру.