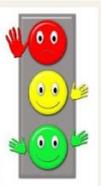


Школа:		
Дата:	ФИО учителя:	
Класс:	Участвовали:	Не участвовали:
Тема урока: Функции. Классы, открытые классы, атрибуты		
Цели обучения, которые достигаются на данном уроке	В этом уроке познакомятся с синтаксисом языка Ruby	
Цели урока	Все учащиеся смогут: <ul style="list-style-type: none"> Узнать об языке <i>Ruby</i> Большинство учащихся смогут: <ul style="list-style-type: none"> научатся находить среду <i>Ruby</i> Некоторые учащиеся смогут: <p><i>Написать первую программу</i></p>	
Критерии оценивания	Грамотность -1балл Оригинальность -1балл Время -1 балл	
Воспитаниеценностей	Воспитание уважения к мнению друг друга, ответственность, коммуникативные способности, критическое мышление	
Предварительныезнания	История Ruby, «Интерпретация» Компиляция и интерпретация, для программистов, Что умеет Ruby? Что не умеет Ruby?	
Межпредметныесвязи	Информатика, математика	
Запланированные этапы урока	Запланированная деятельность на уроке	Ресурсы
Начало урока _5_ мин	Организационный момент. Приветствие учащихся. Создание положительного эмоционального фона. Настрой на рабочий лад. Показываю тему урока и цель урока, озвучиваю критерии для цели урока Деление на подгруппы. Стратегия «Нумерация»	Карточки с цифры
Середина урока _30_ мин	Организационный момент. <i>Определите смысл этих слов, что они из себя представляют:</i> <i>CSS / HTML</i> <i>PHP</i> <i>SQL</i> <i>JavaScript</i> <i>Python</i> <i>Go</i> <i>Java</i> <i>Ruby</i> <i>C++</i> <i>C</i> <ul style="list-style-type: none"> Детям раздаются карточки с вышеуказанными словами, дети в группе определяют что это и где они используются, они могут воспользоваться мобильными средствами или классным 	Карточки для стратегии светофор 

компьютеров на его исполнение некоторые ограничения...



Классы в Руби это уникальные объекты --- каждый из которых является экземпляром класса Class. Когда новый класс создается (обычно используется `class Name ... end`), объект типа Class создается и присваивается одноименной глобальной переменной (в данном случае Name). Когда вызывается `Name.new` для создания нового объекта, запускается метод `new` класса Class. Это можно продемонстрировать перегрузкой метода `new` в классе Class:

```
class Class
  alias oldNew new
  def new(*args)
    print "Создается новый #{self.name}\n"
    oldNew(*args)
  end
end

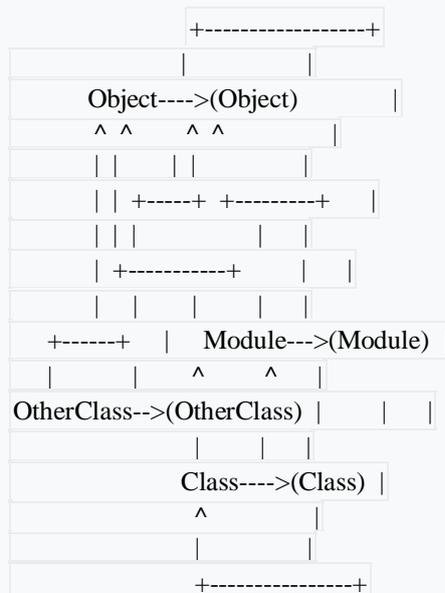
class Name
end

n = Name.new
```

результат:

Создается новый Name

Классы, модули и объекты взаимосвязаны. На следующей диаграмме, вертикальными стрелками обозначено наследование, а круглыми скобками --- метаклассы. Все метаклассы это объекты класса `Class`.



Объявленные атрибуты будут доступны в пределах иерархии наследования, где каждый потомок получает копию атрибутов своих родителей, вместо истинного указателя на него. Это означает,

	<p>что потомок может добавить элементы, для примера, в массив без тех дополнений которые он делит со своим родителем, элементами одного с ним уровня (имеющими общего родителя) или потомками, которые не похожи на правильные атрибуты уровня класса и которые разделены поперек всей иерархии</p> <hr/> <p>Методы класса <u>new</u></p> <p>Методы объекта <u>allocate</u>, <u>inherited</u>, <u>new</u>, <u>superclass</u></p>	
<p>Конец урока _5_ мин</p>	<p>Рефлексия: Метод «Рефлексивные карточки» Было интересно... Я понял, что... У меня получилось... Расскажу дома, что...</p>	<p>«Рефлексивные стикеры»</p> 
<p>Дифференциация – каким образом Вы планируете оказать больше поддержки? Какие задачи Вы планируете поставить перед более способными учащимися?</p>	<p>Оценивание – как Вы планируете проверить уровень усвоения материала учащимися?</p>	<p>Охрана здоровья и соблюдение техники безопасности</p>