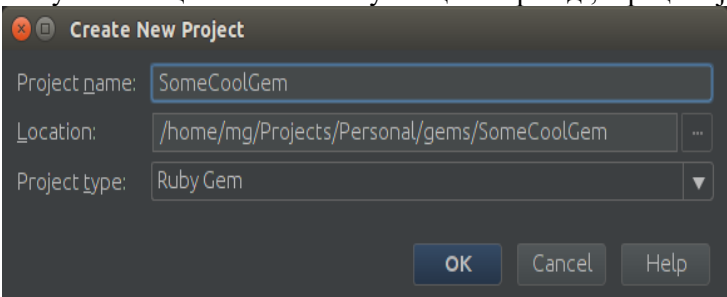
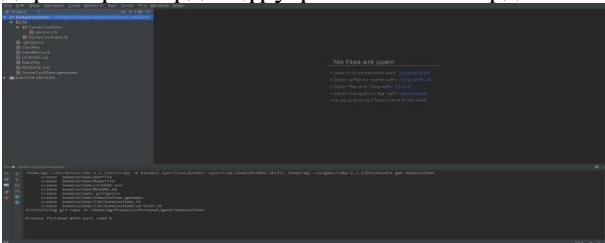


Мектеп:		
Күні:	Мұғалімнің аты – жөні: Сыздықов Р.Ж.	
Сынып:	Қатысқандар саны:	Қатыспағандар саны:
Сабақтың тақырыбы:	Қосымшаны рәсімдеу, геммдермен (модульдермен) жұмыс.	
Осы сабақта қол жеткізілетін оқу мақсаттары	Бағдарламалау тілі кітапханасымен танысу, жобаның прототипін жасау, геммдермен танысу.	
Сабақтың мақсаттары	Барлық оқушылар: Ruby бағдарламалау тілінде жұмыс істеу. Оқушылардың басым бөлігі: кітапханаларды пайдалану. Кейбір оқушылар: жобаларды орындау.	
Бағалау критерийі	Ruby бағдарламалау тілдері туралы біледі. Кітапханаларды, бағдарлама интерфейсін пайдалана алады. Практикада пайдаланады.	
Құндылықтарды дарыту	* Коммуникативтік дағдылар. *Басқа көзқарасқа құрмет көрсету.	
Алдыңғы білімдері	Ruby бағдарламасы. Файлдармен жұмыс. Жүйелік қоршаумен жұмыс.	
Пәнаралық байланыс	Информатика	
Сабақтың жоспарланған кезеңдері	Сабақтағы жоспарланған іс-әрекет	Ресурстар
Сабақтың барысы 5 мин	Ұйымдастыру сәті. Сәлемдесу. Түрлі-түсті стикерлердің көмегімен топтарға бөлу. Топтарға бөлініп, сабаққа дайындалыңыз. Балалар тақтаға назар аударып, тақырыпты анықтайық.	Слайд.
Сабақтың ортасы 30 мин	<p>1-қадам: болванка жасау. RubyMine жаңа жобаны жасау сияқты көрінеді, бірақ Project Type: Ruby Gem.</p>  <p>Бұдан әрі Ruby нұсқасын таңдау болады. Бұл, негізінде, анық және түсіндіруді талап етпейді. Нәтижесінде осындай құрылыммен жобаны аламыз (мен bin және test директорияларды құру үшін галочкаларды алып тастадым).</p>  <p>Барлық құрылымды ретімен қарастырайық. lib-тікелей сіздің қолданбаңыздың барлық кодын (ал, немесе дерлік) сақтайды; .gitignore — нұсқаларды бақылауға түспейтін файлдар мен қалталар тізімін сақтайды (ДБ файлдары, компиленген ассеттер, логтар және т. б.); Gemfile, Gemfile.lock-сіз жұмыс істеу немесе суретті әзірлеу үшін пайдаланатын басқа гемдер тізімі. Бірақ гема жағдайында олар қол жетімді емес, өйткені олар файлда *көрсетіледі.gemspec; LICENSE.txt-пайдалану лицензиясын сақтайды. Менің жағдайда-бұл MIT;</p>	Слайдтар. Интерактивті тақтадағы жұмыс терезелері.

Rakefile-rake таски сақтайды. Интернетте оңай табуға болады;
README, md — қолданба "бет". GitHub және RubyDoc бастапқы беттерінен шығады;

SomeCoolGem.gemspec-негізгі Ақпарат және қосымшаның тәуелділігі сақталған кейбір манифест.

2-қадам: қолданба үшін құрылым жасау

Енді lib директориясымен тікелей жұмыс істейтін боламыз.

Somecoolgem Файлы.rb бастапқы болып табылады. Идея бойынша онда барлық логика сақталуға тиіс, бірақ практика көрсеткендей, модульді құру және қосымшаларды функционалдығы бойынша класқа бөлу, содан кейін оларды осы файлға жүктеуге болады. C++/Java/C# тілдерімен таныс адамдар үшін-бұл файл main әдісі ұқсас нәрсе ().

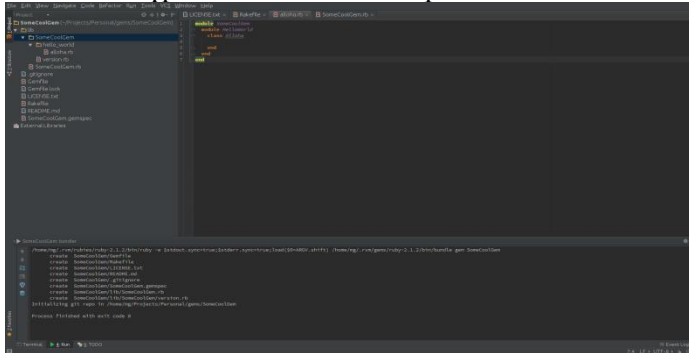
Бірден HelloWorld модулін жасаңыз. Және онда Aloha сlassын жасаймыз.

Өйткені бүкіл гем, шын мәнінде, somecoolgem файлында модуль бар.rb келесі код болады:

```
require "SomeCoolGem/version"
```

```
module SomeCoolGem  
  # Your code goes here...  
end
```

Жобаның құрылымы осылай өзгерді:



Aloha:

```
module SomeCoolGem  
  module HelloWorld  
    class Alloha  
      end  
    end  
  end  
end
```

3-қадам: қажетті функционал жасаймыз:

Aloha класына өріс, конструктор және демонстрация үшін қандай да бір әдістер қосамыз:

```
module SomeCoolGem  
  module HelloWorld  
    class Alloha  
      attr_accessor :name  
      def initialize(name)  
        @name = name  
      end  
      def generate_alloha  
        "Alloha, #{ @name }, from SomeCoolGem!"  
      end  
      def say_alloha  
        puts generate_alloha  
      end  
    end  
  end  
end
```

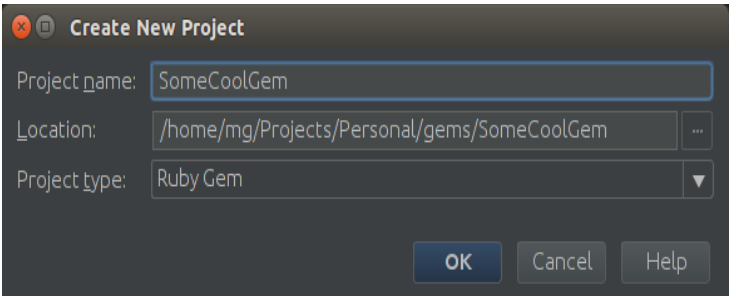
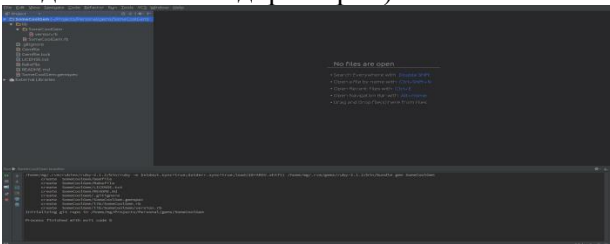
4-қадам: жұмыс қабілетін тексереміз.

	<p>Гем-модуль болғандықтан, оны да модуль ретінде пайдалана аламыз. Тексеру үшін Мен әдетте IRB қолданамын. Міне мысал: 2.1.2 :001 > load 'alloha.rb' => true 2.1.2 :002 > object = SomeCoolGem::HelloWorld::Alloha.new('User') => #<SomeCoolGem::HelloWorld::Alloha:0x0000000e7f950 @name="User"> 2.1.2 :003 > object.generate_alloha => "Alloha, User, from SomeCoolGem!" 2.1.2 :004 > object.say_alloha Alloha, User, from SomeCoolGem! => nil 2.1.2 :005 ></p> <p>Көріп отырғанымыздай, әдетте жобадан айырмашылығы аз. Мысалы, гемді пайдалану кезінде сіздің қолданбаңыз жай ғана салынған модульдер/сыныптар бар модульді қосады және оларды әдеттегі Ruby модульдері/класстары сияқты пайдалануға болады. 5-қадам: басты файлға бәрін қосамыз. Ол үшін біздің бастапқы файлды сәл өзгерту керек. 6-қадам: гем жинау және оның версионизациясы Version файлында.RB біздің гемнің ағымдағы нұсқасы сақталады: module SomeCoolGem VERSION = "0.0.1" end</p> <p>Әрбір босату алдында бұл мәнді өзгерту керек. Бірден айтайын, егер сіз жаңа коммит жасасаңыз, нұсқаны өзгерту міндетті емес. Алайда, егер бағдарлама логикасы өзгерсе, онда босату жақсы. Гем нұсқасын жинау үшін команданы орындау керек: gem build SomeCoolGem.gemspec Жобаның түбірінде SomeCoolGem-0.0.1 файлы пайда болады.gem. Құттықтаймын. Гемнің бірінші нұсқасы жиналды! Бейнеролик көру. Оқушыларды оқыту үдерісіне тарту және сабақ тақырыбына тарту мақсатында бейнероликтерді қарау ұсынылады</p>	<p>https://www.youtube.com/watch?v=NeobYgRhjsQ</p>
--	---	--

<p>Сабақтың соңы. 5 мин</p>	<p>Кahoot. it сайтында тапсырманы орындау.</p>	<p>Кahoot.it сайты</p>
------------------------------------	--	------------------------

<p>Дифференциация – оқушыларға көбірек қолдау көрсетуді қалай жоспарлайсыз? Қабілеті жоғары оқушыларға қандай тапсырмалар қоюды жоспарлап отырсыз?</p>	<p>Бағалау – оқушылардың материалды меңгеру деңгейін тексеру жоспарыңыз?</p>	<p>Денсаулық және қауіпсіздік техникасының сақталуы</p>
<p>1. Қолдау деңгейі бойынша 2.Топтық жұмыстағы рөлі</p>	<p>1. Ұғыну шаблон бойынша 2. Қалыптастыру үшін тапсырманы орындау 3.Рефлексия қорытындысы бойынша тапсырмаларды орындау нәтижесі бойынша кері байланыс ("қол шапалақпен»)</p>	<p>Кабинеттегі жұмыс кезіндегі ҚТ ережесі, психологиялық жайлылық</p>
<p>Сабақ бойынша Рефлексия Сабақтың нақты және қолжетімді мақсаты немесе оқу мақсаттары болды ма? Барлық оқушылар оқу мақсатына жетті ма? Егер оқушылар әлі мақсатқа жетпесе, неге ойлайсыз? Сабақта дифференциация дұрыс жүргізілді?</p>	<p>Сабақты жоспарлауға сабақты ұйымдастырудың белсенді формалары енгізілген:</p>	

Сабақ кезеңінде сіз тиімді пайдаландыңыз ба? Сабақ жоспарынан ауытқу болды ма және неге?		
--	--	--

Школа:		
Дата:	ФИО учителя: Сыздков Р.Ж.	
Класс:	Участвовали:	Не участвовали:
Тема урока:	Оформление приложения, работа с геммами (модулями).	
Цели обучения, которые достигаются на данном уроке	Знакомство с библиотекой языка программирования, сделать прототип проекта, ознакомление с геммами.	
Цели урока	<i>Все учащиеся смогут:</i> выполнить работу на языке программирования Ruby. <i>Большинство учащихся смогут:</i> использовать библиотеки. <i>Некоторые учащиеся смогут:</i> выполнить проекты.	
Критерии оценивания	Знают про языки программирования Ruby. Умеют пользоваться библиотеками, интерфейсом программы. Применяют на практике.	
Воспитание ценностей	*Коммуникативные навыки. *Проявление уважения к другой точке зрения.	
Предварительные знания	Программа Ruby. Работа с файлами. Работа с системным окружением.	
Межпредметные связи	Информатика	
Запланированные этапы урока	Запланированная деятельность на уроке.	Ресурсы
Начало урока 5 мин	Организационный момент Приветствие Поделить на группы с помощи разноцветных стикеров. Разделится по группам,приготовится на урок. Ребята обратите внимание на доску и давайте определим тему.	Слайд.
Середина урока 30 мин	<p>Шаг 1: Создание болванки. В RubyMine это выглядит как создание нового проекта, но с выбором Project Type: Ruby Gem.</p>  <p>Далее будет выбор версии Ruby. Что, в принципе, ясно и не требует объяснения. В итоге получаем проект с такой структурой (я убрал галочки для создания bin и test директорий).</p>  <p>Рассмотрим по порядку всю структуру. lib — непосредственно хранит весь код (ну или почти весь) вашего приложения; .gitignore — хранит список файлов и папок, которые не должны попасть под контроль версий (файлы бд, скомпилированные ассеты, логи и т.п.);</p>	Слайды. Рабочие окна на интерактивной доске.

Gemfile, Gemfile.lock — список других гемов, которые вы используете для работы или при разработке изображения. Но в случае гема они недоступны, т.к. они указываются в файлу *.gemspec;

LICENSE.txt — хранит лицензию использования. В моем случае — это MIT;

Rakefile — хранит Rake задачи. Подробнее можно легко найти в интернете;

README.md — «лицо» вашего приложения. Выводится на стартовых страницах GitHub и RubyDoc;

SomeCoolGem.gemspec — некоторый манифест, в котором хранится основная информация и зависимости приложения.

Шаг 2: создание структуры для работы приложения

Теперь будем работать непосредственно с директорией lib.

Файл SomeCoolGem.rb является стартовым. По идее в нем должна храниться вся логика, но, как показывает практика, гораздо целесообразней создать модуль и разделить приложение на классы по функциональности, а затем просто подгружать их в этот файл. Для тех, кто знаком с языками C++/Java/C# — этот файл чем-то сродним методу main().

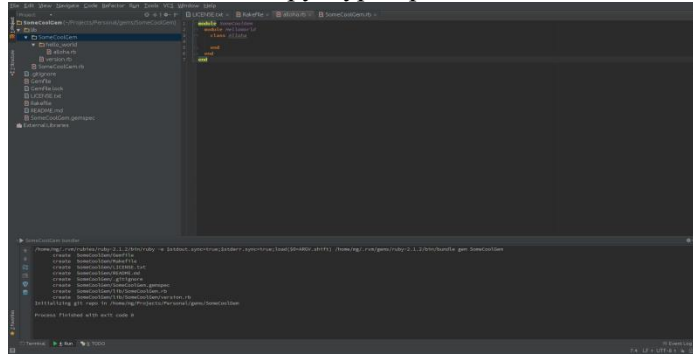
Сразу создадим модуль HelloWorld. И в нем создадим класс Aloha.

Так как весь гем, по сути, есть модуль то уже в файле SomeCoolGem.rb будет следующий код:

```
require "SomeCoolGem/version"
```

```
module SomeCoolGem
  # Your code goes here...
end
```

Вот так изменилась структура проекта:



Вот класс Aloha:

```
module SomeCoolGem
  module HelloWorld
    class Aloha
    end
  end
end
```

Шаг 3: Делаем нужный функционал

Добавим в класс Aloha поле, конструктор и какие-нибудь методы для демонстрации:

```
module SomeCoolGem
  module HelloWorld
    class Aloha
      attr_accessor :name
      def initialize(name)
        @name = name
      end
    end
  end
end
```

```

def generate_alloha
  "Alloha, #{@name}, from SomeCoolGem!"
end
def say_alloha
  puts generate_alloha
end
end
end
end

```

Шаг 4: Проверяем работоспособность

Так как гем — это модуль, то использовать мы его можем тоже как модуль. Для проверки я обычно использую **IRB**.

Вот пример:

```

2.1.2 :001 > load 'alloha.rb'
=> true
2.1.2 :002 > object = SomeCoolGem::HelloWorld::Alloha.new('User')
=> #<SomeCoolGem::HelloWorld::Alloha:0x00000000e7f950
@name="User">
2.1.2 :003 > object.generate_alloha
=> "Alloha, User, from SomeCoolGem!"
2.1.2 :004 > object.say_alloha
Alloha, User, from SomeCoolGem!
=> nil
2.1.2 :005 >

```

Как видим, мало чем отличается от работы обычно проекта. Грубо говоря, при использовании гема ваше приложение просто подключает модуль с вложенными модулями/классами и вы можете использовать их как и обычные модули/классы Ruby.

Шаг 5: Подключаем все к главному файлу

Для этого нужно немного изменить наш стартовый файл SomeCoolGem.rb:

```

require "SomeCoolGem/version"
module SomeCoolGem
  require 'SomeCoolGem/hello_world/alloha'
end

```

Теперь при указании нашего гема в Gemfile приложения мы сможем использовать его как обычный модуль Ruby.

Шаг 6: Сборка гема и его версионизация

В файле version.rb хранится текущая версия нашего гема:

```

module SomeCoolGem
  VERSION = "0.0.1"
end

```

Перед каждым релизом нужно будет менять это значение.

Сразу скажу, что если вы делаете новый коммит, то не обязательно менять версию. Однако если меняется логика программы, то лучше сделать релиз.

Чтобы собрать версию гема, вам нужно выполнить команду:

```
gem build SomeCoolGem.gemspec
```

В корне проекта появится файл SomeCoolGem-0.0.1.gem.

Поздравляю. Первая версия гема собрана!

Просмотр видеоролика.

С целью вовлечения учащихся в процесс обучения и подведения к теме урока предлагается посмотреть видеоролик

<https://www.youtube.com/watch?v=NeobYgRhjsQ>

Конец урока
5 мин

Выполнение задания на сайте Kahoot.it

сайт Kahoot.it

Дифференциация – каким образом Вы планируете оказать больше поддержки? Какие задачи Вы планируете поставить перед более способными учащимися?

Оценивание – как Вы планируете проверить уровень усвоения

Охрана здоровья и соблюдение техники безопасности

	материала учащимися?	
<p>1. По уровню поддержки 2. По роли в групповой работе</p>	<p>1. Самооценивание по шаблону 2. Выполнение задания для ФО 3. Обратная связь по итогам выполнения заданий, по итогам рефлексии («аплодисменты»)</p>	<p>Правила ТБ при работе в кабинете, Психологический комфорт</p>
<p>Рефлексия по уроку Была ли реальной и доступной цель урока или учебные цели? Все ли учащиеся достигли цели обучения? Если ученики еще не достигли цели, как вы думаете, почему? Правильно проводилась дифференциация на уроке? Эффективно ли использовали вы время во время этапов урока? Были ли отклонения от плана урока, и почему?</p>	<p>В планирование урока включены активные формы организации урока:</p>	