



## 06. Объектное программирование на Ruby

Практические задания.

**Цель:** научиться разрабатывать программы с применением классов, методов и объектов.

**Задача:** Разработать программу для управления метеостанцией.

№ 06.0

1. Запустите редактор **Geany** из раздела «Программирование» в главном меню.
2. Создайте в редакторе **Geany** новый файл `meteo_station.rb`. Сохраните программу `meteo_station.rb` в рабочем каталоге `~/projects/`.
3. Пошагово вводите в него дополняемые варианты текста программы. Изменения в тексте выделены красным цветом. Не забывайте сохранять изменения.
4. После каждого изменения запускайте программу на выполнение из раздела меню «Сборка», пункт «Execute» (выполнить), проверяйте её работу по выводимым сообщениям в открывшемся терминальном окне. Закрывайте терминальное окно после очередного прогона программы.

№ 06.1

1. Начальный вариант текста программы только с описанием класса и одним проверочным методом:

```
#!/usr/bin/ruby

class MeteoStation # v.0
  def start
    printf("%s Начало работы\n",
           Time.now.strftime("%Y-%m-%d %H:%M:%S"))
  end
end

meteo = MeteoStation.new
meteo.start
```

2. Дополненная программа с отдельным методом протоколирования.

```
class MeteoStation # v.1
  def start
    log("Начало работы")
  end

  def log(message)
    timestamp = Time.now.strftime("%Y-%m-%d %H:%M:%S")
    printf("%s %s\n", timestamp, message)
  end
end

meteo = MeteoStation.new
meteo.start
```

3. Изменённая программа с добавленными атрибутами «температура» и «влажность» и методом инициализации объекта.

```
require "dht_sensor" # подключить библиотеку для DHTxx

class MeteoStation # v.2
```

```

attr :sensor_th          # объявить атрибут «датчик»
attr :temperature        # объявить атрибут «температура»
attr :temperature, :humidity # объявить атрибуты «влажность»

def initialize
  @sensor_th = DHT11.new(23)          # создать объект класса DHT11
  @sensor_th.read_data                # считать показания датчика
  @temperature = @sensor_th.celsius   # записать значение в атрибут
  @humidity    = @sensor_th.humidity # записать значение в атрибут
end

def start
  log("Начало работы")
  log(@temperature.to_s+" "+@humidity.to_s)
end

def log(message)
  timestamp = Time.now.strftime("%Y-%m-%d %H:%M:%S")
  printf("%s %s\n", timestamp, message)
end
end

```

```

meteo = MeteoStation.new
meteo.start

```

4. Улучшенная версия программы с отдельным методом обновления показаний, методом преобразования в строку и циклом вывода в протокол.

```

class MeteoStation # v.3
  attr :sensor_th
  attr :temperature
  attr :temperature, :humidity

  def initialize
    @sensor_th = DHT11.new(23)
    self.measure
  end

  def measure
    @sensor_th.read_data
    @temperature = @sensor_th.celsius
    @humidity    = @sensor_th.humidity
    return [@temperature, @humidity]
  end

  def values_as_string
    (temperature, humidity) = measure
    temperature.to_s+" "+humidity.to_s
  end

  def start
    log("Начало работы")
    log(values_as_string)
  end

  def log(message)
    timestamp = Time.now.strftime("%Y-%m-%d %H:%M:%S")
    printf("%s %s\n", timestamp, message)
  end
end

```

```

meteo = MeteoStation.new
meteo.start

```

5. Обновлённая версия программы с циклом измерений и вывода значений в протокол. Добавьте в подходящее место в программе цикл, где текущие показания будут выводиться в файл с определённой периодичностью.

*Не лучше ли для этого сделать отдельный метод?*

*Как удобнее задать периодичность измерений?*

6. Как можно ещё улучшить программу?

Окончательный вариант программы можно посмотреть в файле  
`~/CodeClub-IoT/github/samples/meteo_station_5.rb`.