Автоматное программирование

Кирилл Мокевнин
Онлайн-школа программирования Хекслет / сооснователь

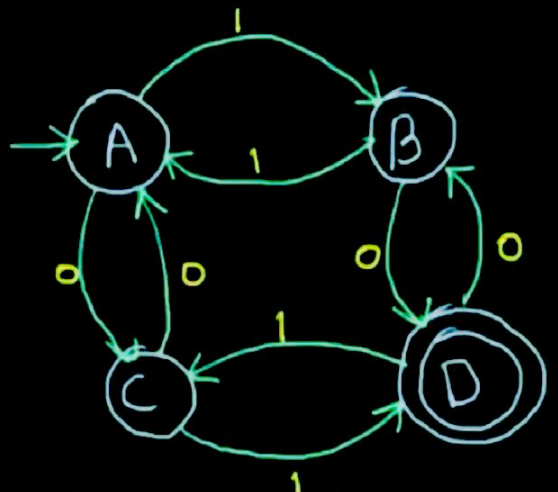
Хекслет

Об авторе:

- С 2007 года в разработке
- С 2013 сооснователь Хекслета
- Направления: Devops, Frontend, Backend

Хекслет

FSM (Finite Automata)



$(Q, \Sigma, q_0, F, \delta)$

Q = set of all states

Σ = inputs

q_0 = start state / initial state

F = set of final states.

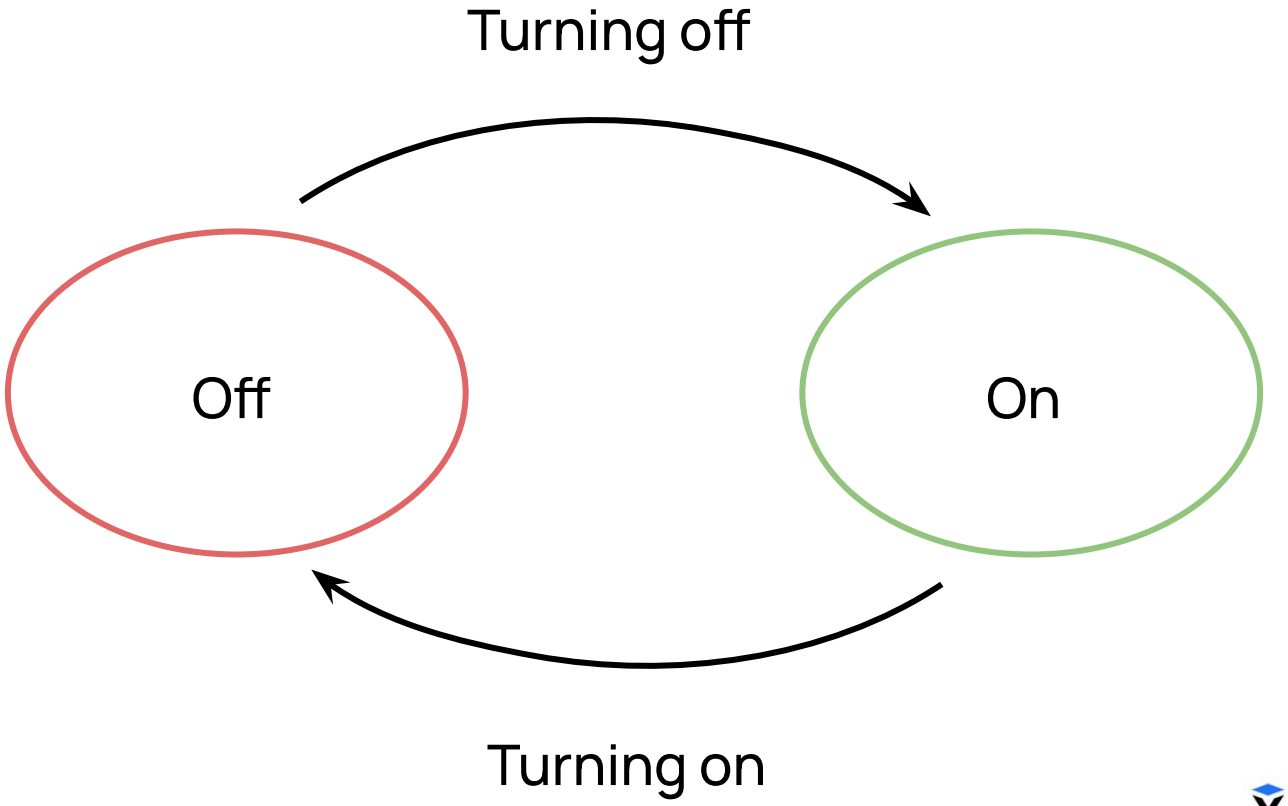
δ = transition function from $Q \times \Sigma \to Q$

03 TOC

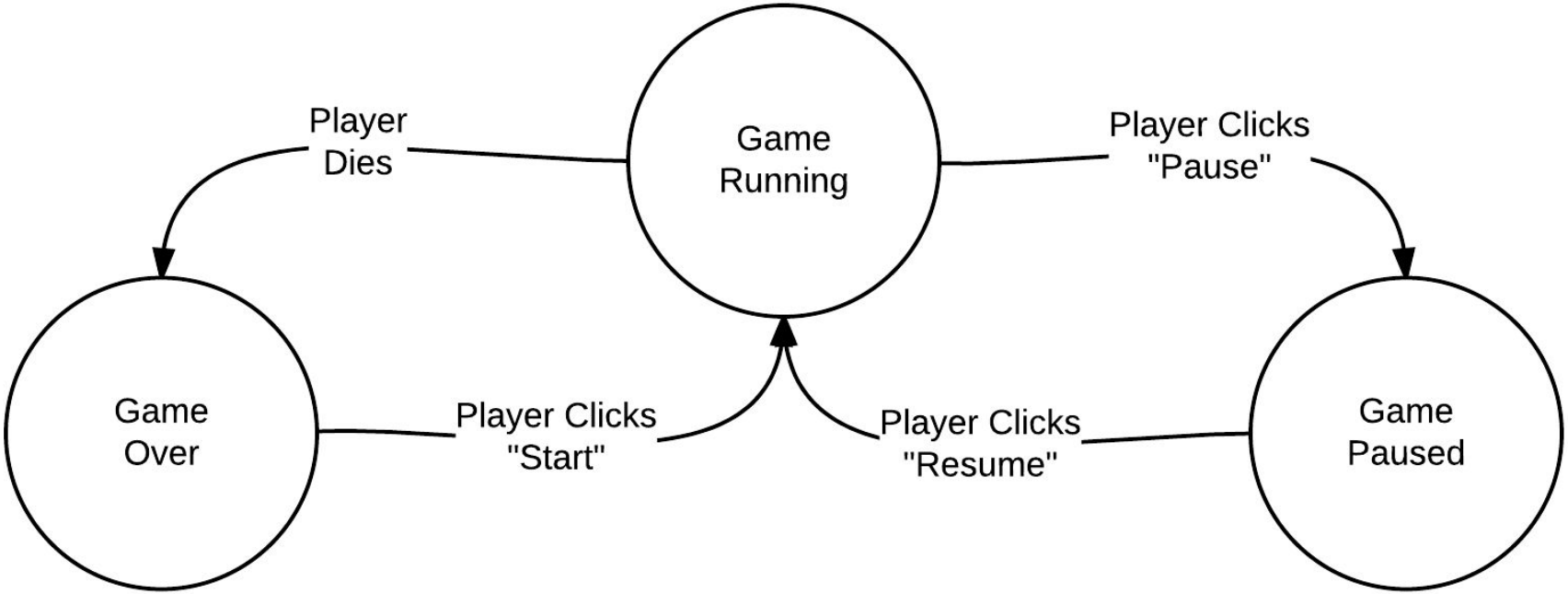
Хекслет

Автоматное программирование != Паттерн State

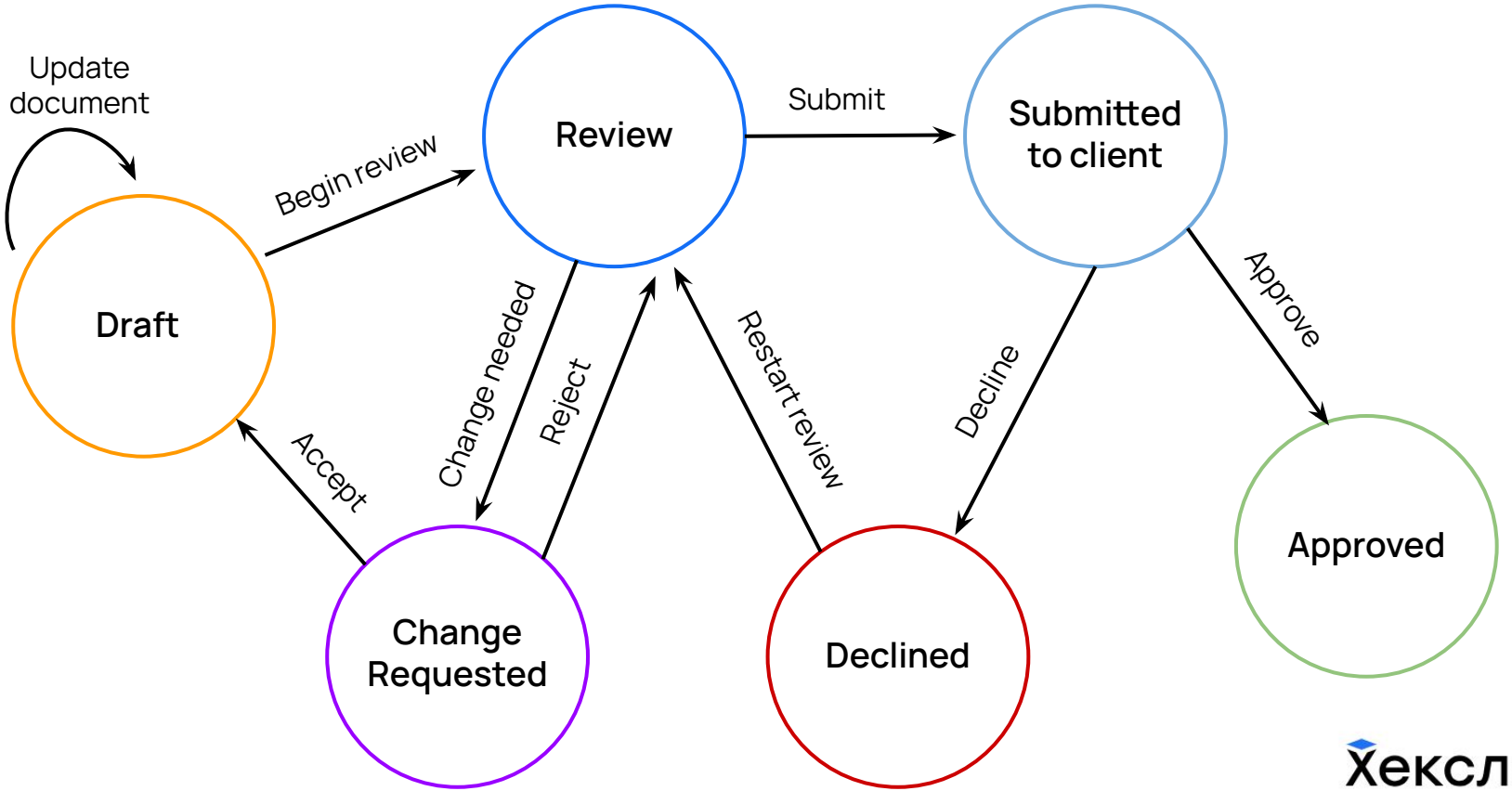
Хекслет



Хекслет


Player
Dies
Game
Running
Player Clicks
"Pause"
Game
Over
Player Clicks
"Start"
Player Clicks
"Resume"
Game
Paused

Хекслет


Update document
Draft
Begin review
Review
Submit
Submitted to client
Approve
Approved
Decline
Declined
Restart review
Reject
Change needed
Accept
Change Requested

Хекслет

Программирование с явно выделенным состоянием

Хекслет

Неявное состояние

```
1
2 user = User.find(params[:id])
3
4 if !user.token?
5   redirect_to root_path
6 end
```

Хекслет

Явное состояние

```
1
2 user = User.find(params[:id])
3
4 if !user.waiting_confirmation?
5   redirect_to root_path
6 end
```

Хекслет

База данных

- Текстовое поле даже если два состояния
- На каждый автомат свое поле

```
CREATE TABLE public.users (
    id bigint NOT NULL,
    state character varying(255),
    stripe_state character varying(255),
    email character varying(255),
    created_at timestamp without time zone,
    updated_at timestamp without time zone,
);
```

Хекслет

Реализация

```ruby
class Job
  include AASM

  aasm do
    state :sleeping, initial: true
    state :running, :cleaning

    event :run do
      transitions from: :sleeping, to: :running
    end

    event :clean do
      transitions from: :running, to: :cleaning
    end

    event :sleep do
      transitions from: [:running, :cleaning], to: :sleeping
    end
  end

end
```

```ruby
job = Job.new
job.sleeping? # => true
job.may_run?  # => true
job.run
job.running?  # => true
job.sleeping? # => false
job.may_run?  # => false
job.run       # => raises AASM::InvalidTransition
```

Хекслет

Антипаттерн (true/false)

- is_sleeping
- is_cleaning
- is_running

Может ли измениться состояние без события?

Хекслет

```ruby
state_machine initial: :created do
  state :created
  state :generating
  state :generated
  state :uploading
  state :uploaded do
    validates :key, presence: true
  end
  state :failed

  event :generate do
    transition %i[created generating] => :generating
  end

  event :mark_as_generated do
    transition %i[generating generated] => :generated
  end

  event :upload do
    transition %i[created generated] => :uploading
  end

  event :mark_as_uploaded do
    transition [:uploading] => :uploaded
  end

  event :mark_as_failed do
    transition %i[created generating uploading] => :faile
  end
```

```ruby
class InstanceUploadService
  class << self
    def upload(upload)
      instance = upload.instance

      api = EvaluatorResolver.api(upload.instance_run.server_name)

      if upload.can_upload?
        upload.upload!
        if api.upload_code(upload)
          upload.mark_as_uploaded!
          instance.last_upload_at = Time.current
          instance.save!
        else
          upload.mark_as_failed!
        end
      end
    end

    def remove_upload(upload)
      s3 = ApplicationContainer[:s3_resource]

      if upload.uploaded?
        bucket = s3.bucket(configus.code_loader.bucket_name)
        s3_object = bucket.object(upload.key)
        s3_object.delete
      end

      upload.destroy!
    end

  end
end
```

Хекслет

Конечные автоматы для всех

- Автоматы — естественный способ описания процессов
- Код — реализация переходов в автоматах
- Сложность системы определяется количеством автоматов
- Явные автоматы значительно повышают понятность кода

Хекслет

Состояние или атрибут?

- Успешность оплаты
- Цвет текста
- Звонок
- Категория продукта

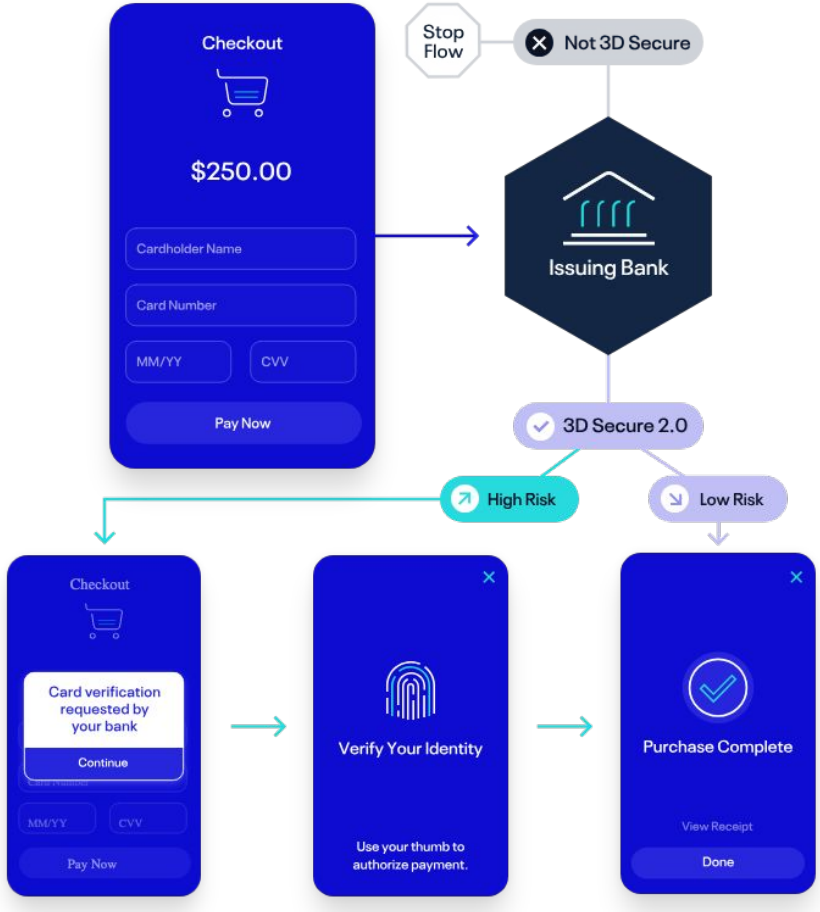
Хекслет

Процессы и автоматы

- Новый
- Зарегистрированный (ожидает подтверждение почты)
- Активный
- Забаненный
- Удаленный

Хекслет

Зависимые автоматы



Хекслет

Библиотеки

- Java: spring-statemachine
- PHP: laravel-model-states
- Python: django-fsm
- Frontend: XState
- GO: looplab/fsm

Хекслет

Ключевые идеи

- Смотреть на происходящие процессы как на конечные автоматы
- Завязываться в коде на явно выделенное состояние, а не косвенные признаки
- Использовать готовые библиотеки для описания, валидации и связи с базой

Хекслет

Кирилл Мокевнин

соосновательХекслета

mokevnin.github.io

Хекслет