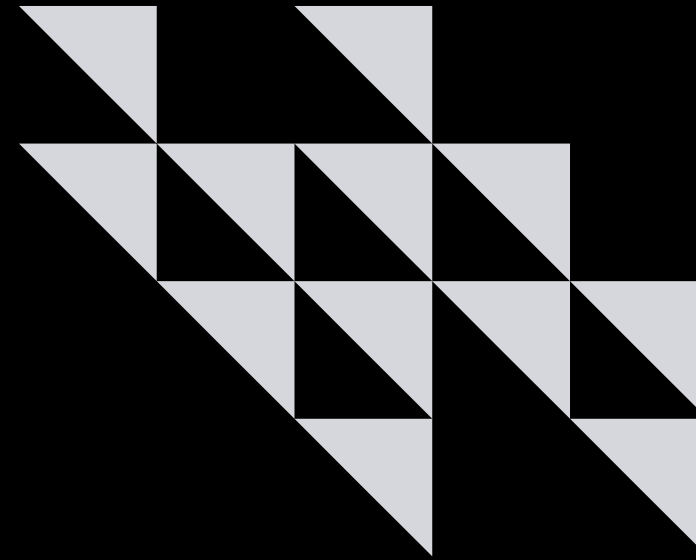


Шаблонний метод



Ruby

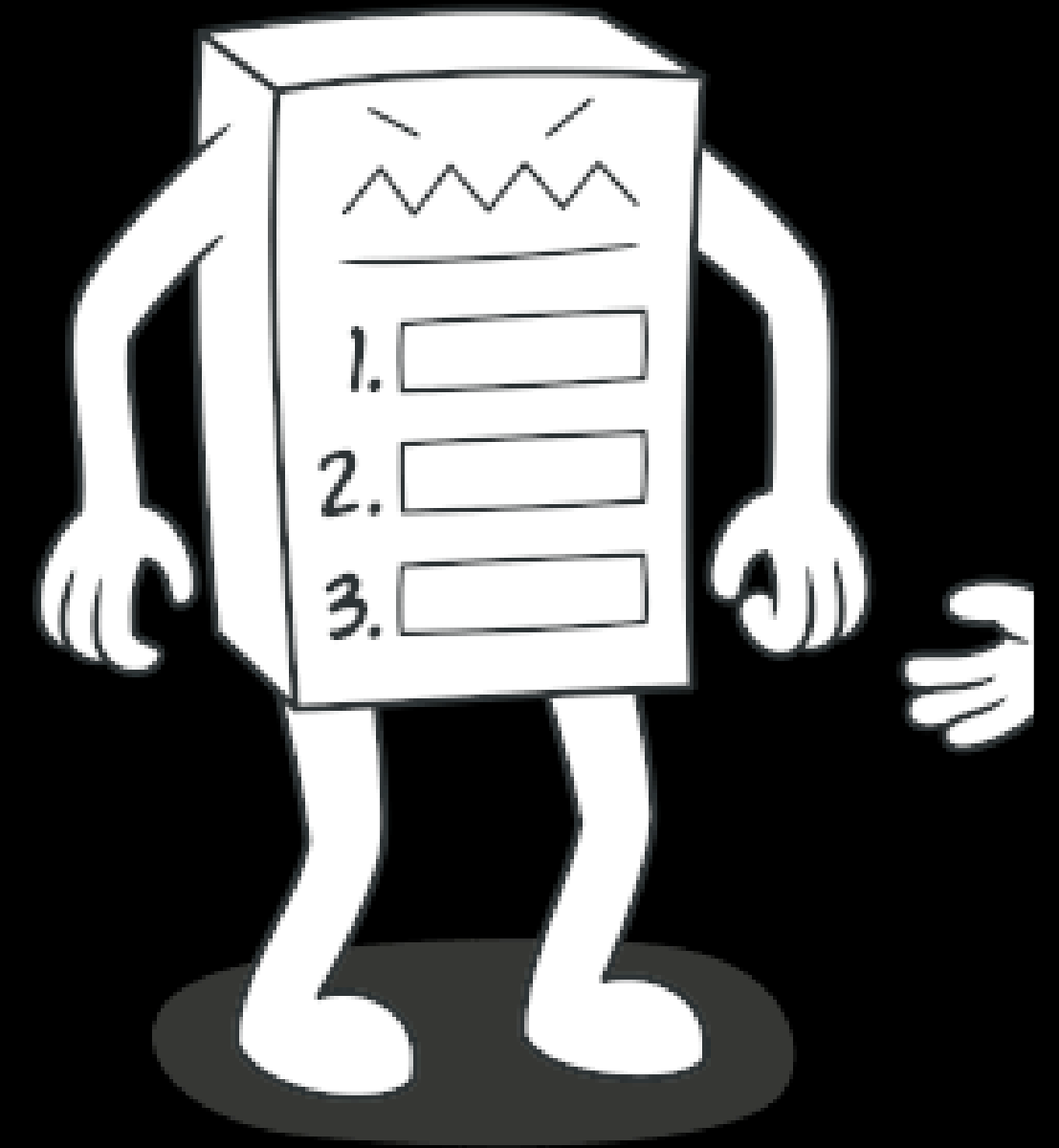
Що таке шаблонний метод?

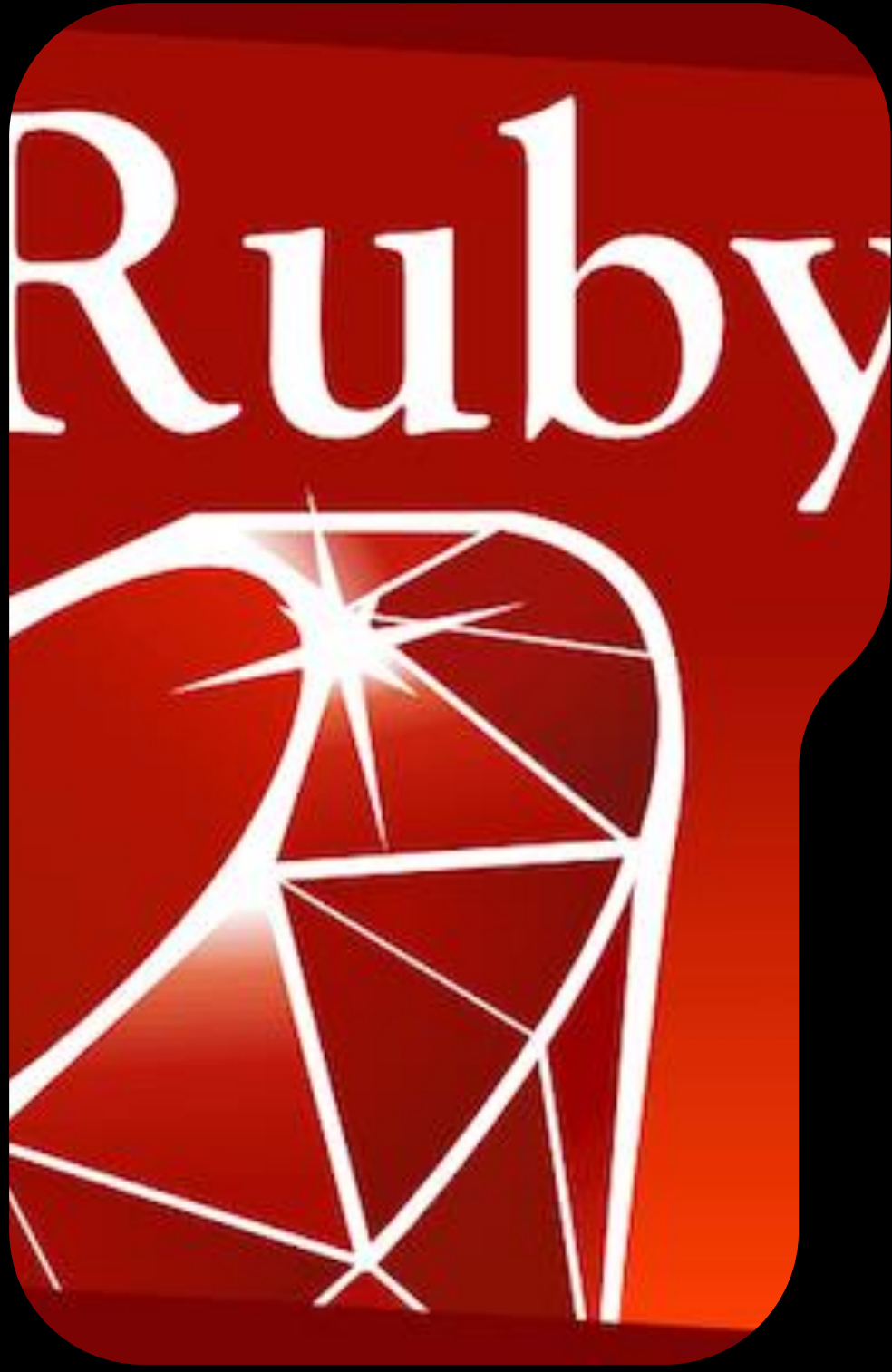


Шаблонний метод — це патерн проектування, який дозволяє визначити загальний алгоритм, делегуючи реалізацію деяких кроків підкласам. Цей патерн належить до категорії поведінкових патернів.

Основні переваги використання патерну:

- Спрощує підтримку коду: Основна логіка алгоритму залишається в базовому класі, зменшуючи дублювання.
- Зменшує дублювання коду: Спільні частини алгоритму реалізуються лише один раз.
- Сприяє узгодженості: Забезпечує однаковий порядок виконання в різних реалізаціях.





Опис патерну

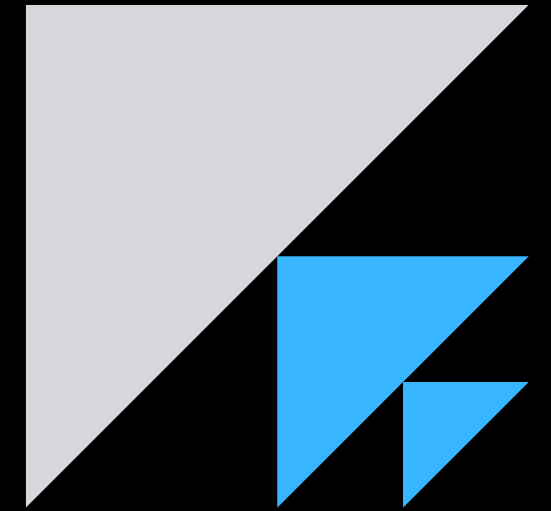
Принцип роботи шаблонного методу:

Шаблонний метод описує загальний алгоритм у базовому класі, але конкретні реалізації кроків алгоритму визначаються в підкласах. Це дозволяє підкласам змінювати певні деталі, не порушуючи загальну структуру.

Коли використовувати:

Використовується, коли є кілька варіантів виконання алгоритму з однаковими основними кроками, але з різними деталями, що реалізуються в підкласах.

ОСНОВНІ КОМПОНЕНТИ



1. Абстрактний клас:

- Містить загальний метод, що реалізує алгоритм. Цей метод викликає інші методи, реалізацію яких залишено для підкласів.

1. Конкретні реалізації:

- Підкласи, які успадковують абстрактний клас та реалізують специфічні кроки алгоритму, такі як налаштування добавок.



Код шаблонного метода



```
class CoffeeTemplate
  def prepare_recipe
    boil_water
    brew_coffee_grounds
    pour_in_cup
    add_condiments
  end

  def boil_water
    puts "Boiling water"
  end

  def pour_in_cup
    puts "Pouring into cup"
  end

  def add_condiments
    raise NotImplementedError, 'You must implement
    add_condiments method'
  end

  def brew_coffee_grounds
    puts "Dripping coffee through filter"
  end
end
```

Конкретний клас для кави

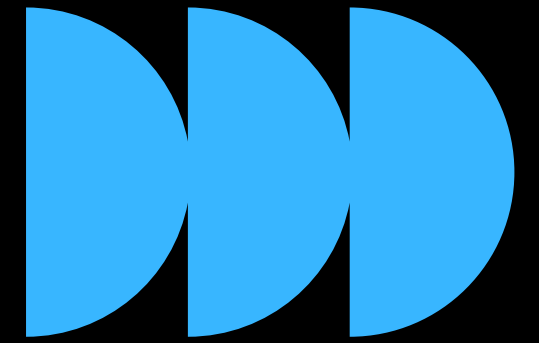
```
class Coffee < CoffeeTemplate
  def add_condiments
    puts "Adding sugar and milk"
  end
end
```

Конкретний клас для чаю

```
class Tea < CoffeeTemplate
  def add_condiments
    puts "Adding lemon"
  end

  def brew_coffee_grounds
    puts "Steeping tea leaves"
  end
end
```

Приклад використання



```
coffee = Coffee.new  
coffee.prepare_recipe
```

```
tea = Tea.new  
tea.prepare_recipe
```

Вивід програми



Boiling water
Dripping coffee through filter
Pouring into cup
Adding sugar and milk

Boiling water
Steeping tea leaves
Pouring into cup
Adding lemon

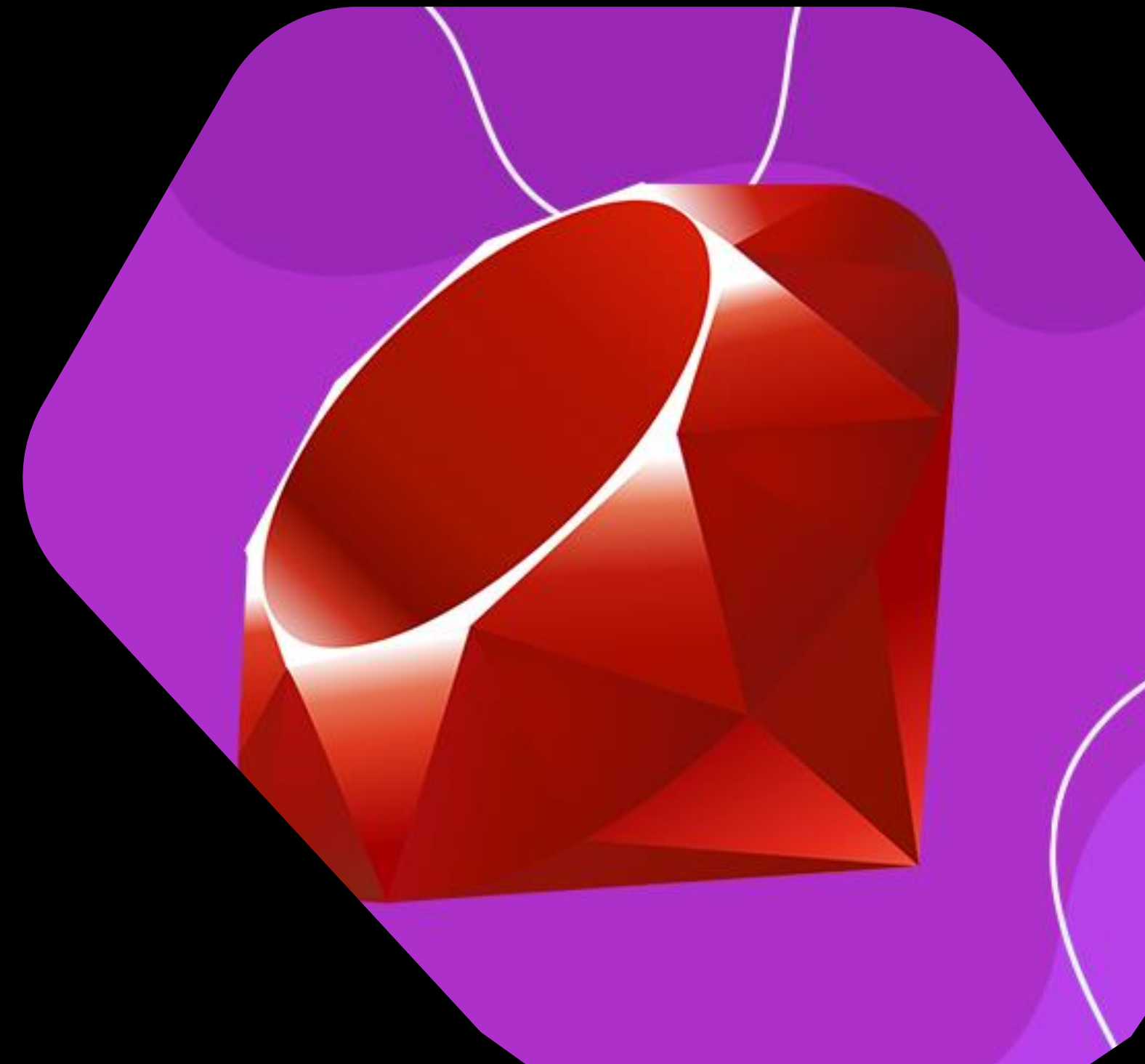
Висновки

1. Переваги використання шаблонного методу:

- Уніфікація коду: забезпечує однаковий підхід до реалізації подібних алгоритмів.
- Легкість у модифікації та тестуванні: спрощує внесення змін до алгоритму, оскільки основна логіка залишається в базовому класі.

1. Як це допомагає в програмуванні:

- Шаблонний метод підвищує читабельність та підтримуваність коду, що є важливим фактором у великих проектах.



Приклади застосування

Сфери використання шаблонного методу:

- Фреймворки: використовується в багатьох фреймворках для обробки запитів, де основна логіка обробки однакова, але конкретні дії відрізняються.
- Ігри: у іграх, де алгоритми руху чи взаємодії з об'єктами можуть бути схожими, але мають свої особливості.
- Бізнес-логіка: у програмних системах, де різні бізнес-процеси мають загальні етапи, але різні кінцеві дії.