Теоретичні питання

1. Яка роль методу Thread#value?

2. Що таке "змагання" (race condition)? Як цього уникнути?

3. Як можна зберегти блок в змінну? Наведіть приклад.

4. Що відбувається, якщо у лямбду передати неправильну кількість аргументів?

Практичні завдання

1. Створіть клас Фігура із методом для обчислення площі. Реалізуйте наслідування для класів Квадрат та Трикутник.

2. Реалізуйте перевантаження оператора + для об'єднання двох об'єктів класу Словник.

1. Яка роль методу Thread#value?

- **Метод Thread#value** використовується для отримання результату виконання потоку (Thread) після його завершення. Якщо потік ще не завершився, метод чекає, доки це станеться, і потім повертає результат.

2. Що таке "змагання" (race condition)? Як цього уникнути?

- **"Змагання" (race condition)** виникає, коли кілька потоків або процесів одночасно взаємодіють з одним і тим самим ресурсом, що може призводити до непередбачуваних результатів.
- **Як уникнути:**
  - Використовуйте **м'ютекси** (mutex).
  - Використовуйте **монітори** або **синхронізацію потоків**.
  - Доступ до ресурсу має бути атомарним.

3. Як можна зберегти блок у змінну? Наведіть приклад.

- Блок можна зберегти у змінну, використовуючи об'єкт Proc або lambda.

4. Що відбувається, якщо у лямбду передати неправильну кількість аргументів?

- Лямбда викликає помилку ArgumentError, якщо кількість переданих аргументів не відповідає оголошеним.

Практичні завдання

1. Створіть клас Фігура із методом для обчислення площі. Реалізуйте наслідування для класів Квадрат та Трикутник.

```ruby
class Figure
  def area
    raise NotImplementedError, "Метод площа має бути перевизначений"
  end
end

class Square < Figure
  def initialize(side)
    @side = side
  end

  def area
    @side ** 2
  end
end

class Triangle < Figure
  def initialize(base, height)
    @base = base
    @height = height
  end

  def area
    0.5 * @base * @height
  end
end

square = Square.new(4)
puts square.area # 16

triangle = Triangle.new(3, 6)
puts triangle.area # 9
```

2. Реалізуйте перевантаження оператора + для об'єднання двох об'єктів класу Словник.

```ruby
class Словник
  attr_accessor :дані

  def initialize(дані = {})
    @дані = дані
  end

  def +(other)
    Словник.new(@дані.merge(other.дані))
  end
end

# Використання:
слово1 = Словник.new({ "apple" => "яблуко", "dog" => "пес" })
слово2 = Словник.new({ "cat" => "кіт", "apple" => "яблуко" })

результат = слово1 + слово2
puts результат.дані
# {"apple"=>"яблуко", "dog"=>"пес", "cat"=>"кіт"}
```

```
{"apple"=>"яблуко", "dog"=>"пес", "cat"=>"кіт"}

Process finished with exit code 0
```