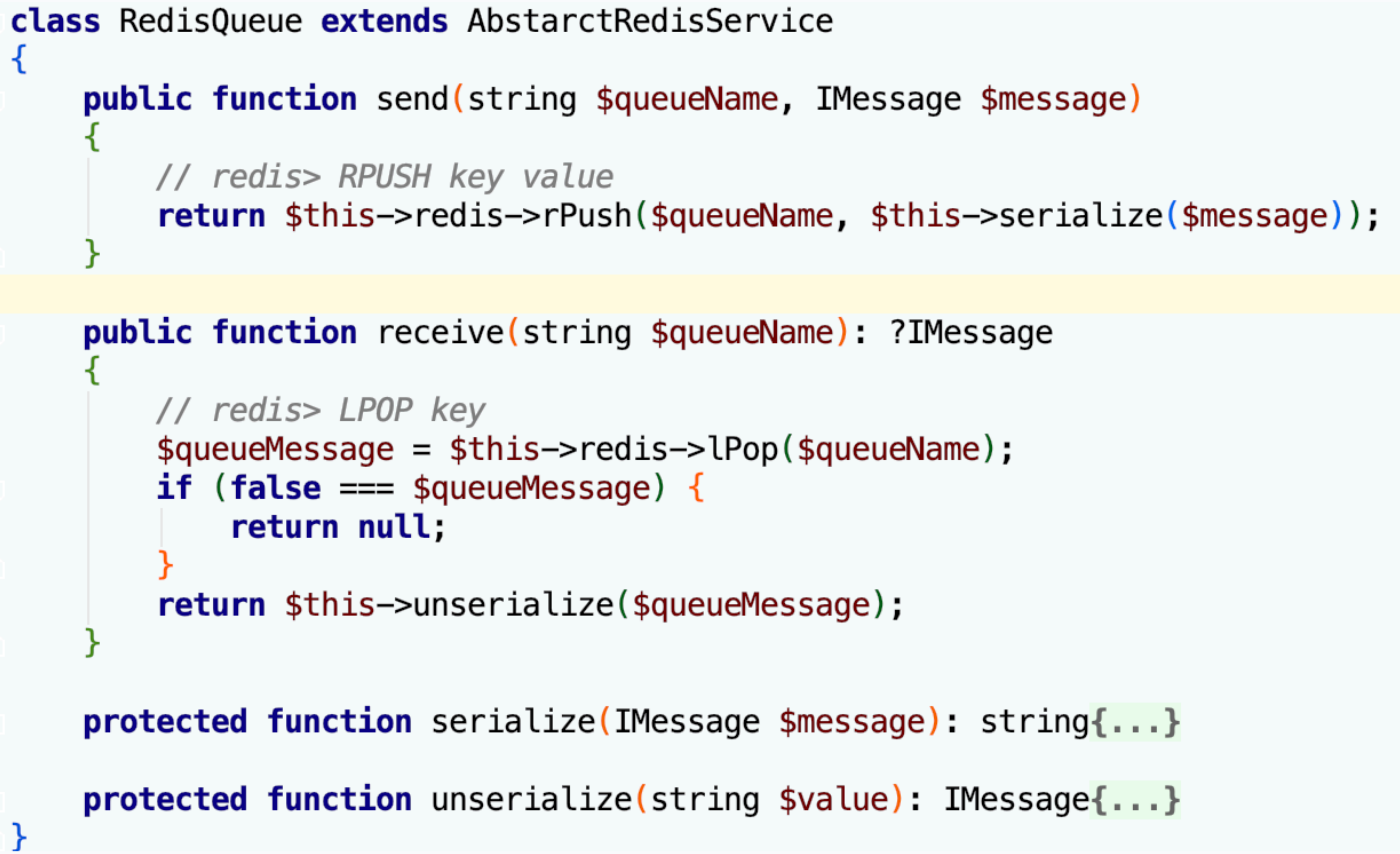
Классическая очередь



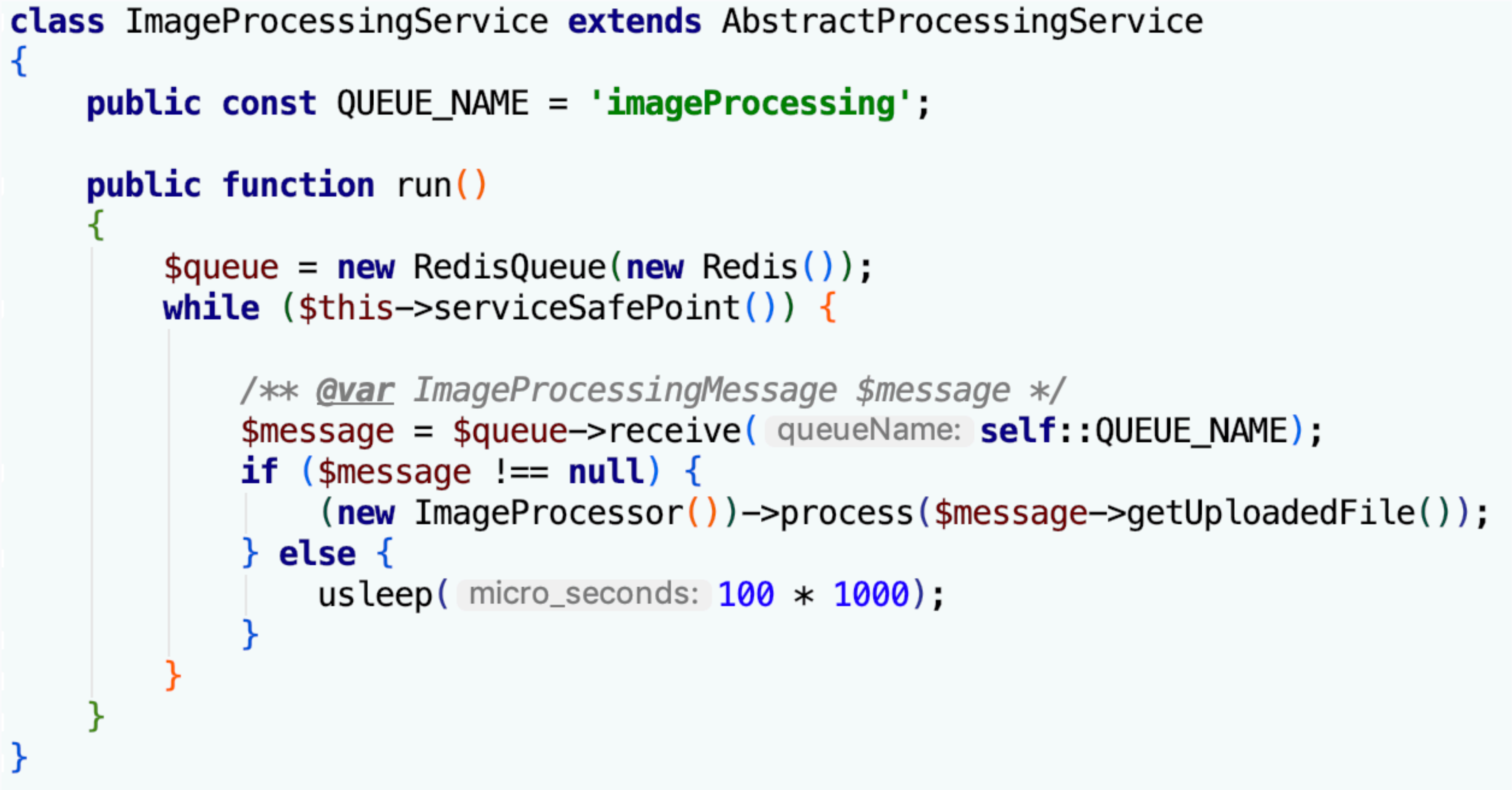
```php
class RedisQueue extends AbstarctRedisService
{
    public function send(string $queueName, IMessage $message)
    {
        // redis> RPUSH key value
        return $this->redis->rPush($queueName, $this->serialize($message));
    }

    public function receive(string $queueName): ?IMessage
    {
        // redis> LPOP key
        $queueMessage = $this->redis->lPop($queueName);
        if (false === $queueMessage) {
            return null;
        }
        return $this->unserialize($queueMessage);
    }

    protected function serialize(IMessage $message): string{...}

    protected function unserialize(string $value): IMessage{...}
}
```

Классическая очередь



```
class ImageProcessingService extends AbstractProcessingService
{
    public const QUEUE_NAME = 'imageProcessing';

    public function run()
    {
        $queue = new RedisQueue(new Redis());
        while ($this->serviceSafePoint()) {

            /** @var ImageProcessingMessage $message */
            $message = $queue->receive( queueName: self::QUEUE_NAME);
            if ($message !== null) {
                (new ImageProcessor())->process($message->getUploadedFile());
            } else {
                usleep( micro_seconds: 100 * 1000);
            }
        }
    }
}
```