

# 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);
            }
        }
    }
}
```

# Инерциальные очереди

```
public function actionUpdate(UpdateRequest $request)
{
    $user = new User($request->getUserId());
    $user->updateData($request->getData());

    $queue = new RedisQueue(new Redis());
    $queue->send( queueName: 'userAnalytics', new UserAnalyticsMessage($request->getUserId()));
}
```