```
type Result<TResult, TError> =
  Success<TResult> |
  Failure<TError>;
```

```
type Success<T> = {
  kind: "successfulResult";
  value: T;
};
```

```
type Failure<T> = {
  kind: "failedResult";
  error: T;
};
```

```ts
type Result<TResult, TError> =
  Success<TResult> |
  Failure<TError>;


type Success<T> = {
  kind: "successfulResult";
  value: T;
};


type Failure<T> = {
  kind: "failedResult";
  error: T;
};
```

```
type OrderResult =
  Result<ProcessedOrder | OrderAction,
  OrderProcessingError>;

function processOrder(
  order: Order,
  options: OrderOptions): OrderResult {
```