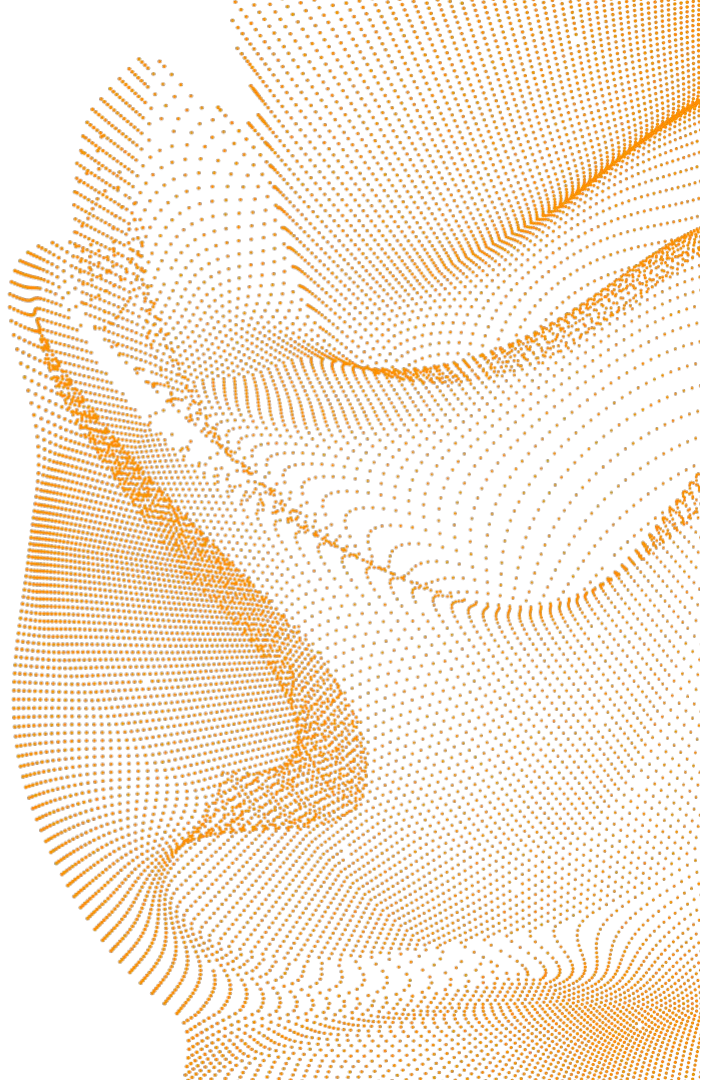
ОРЕНБУРГ | 12 — 13 СЕНТЯБРЯ | 2025

ПРОФЕССИОНАЛЬНАЯ МЕЖРЕГИОНАЛЬНАЯ

IT-КОНФЕРЕНЦИЯ MERGE



merge

ОРЕНБУРГ | 12 — 13 СЕНТЯБРЯ | 2025

EVRONE.RU

FP guru

Павел
Аргентов

ФП и бизнес-логика

ЧТО ОБЩЕГО У ПАРОВОЗА И АЛГЕБРЫ?

Неудачная метафора подобна котенку с дверцей (anonim., c. 2000 AD)

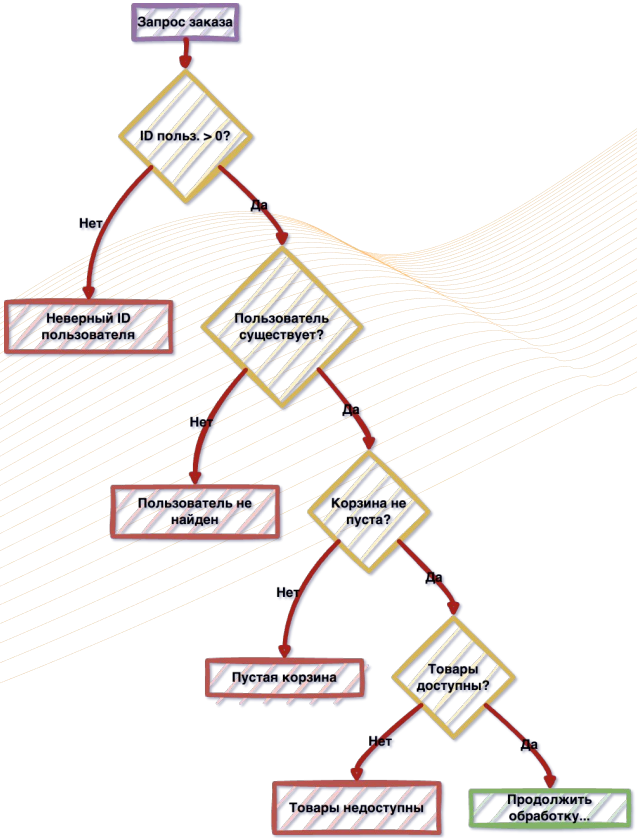
merge

EVRONE
FP guru

Бизнес-транзакция

ТРАДИЦИОННЫЕ ЦЕННОСТИ

1. **Скрытый поток управления**: Исключения прерывают выполнение непредсказуемо
2. **Фокус на ошибках**: Код больше занят обработкой ошибок, чем бизнес-логикой
3. **Хрупкость системы**: Легко забыть обработать какой-то случай
4. **Сложность тестирования**: Трудно протестировать все возможные пути выполнения
5. **Потеря контекста**: При исключении теряется информация о промежуточных состояниях



EVRONE
FP guru

Бизнес-транзакция

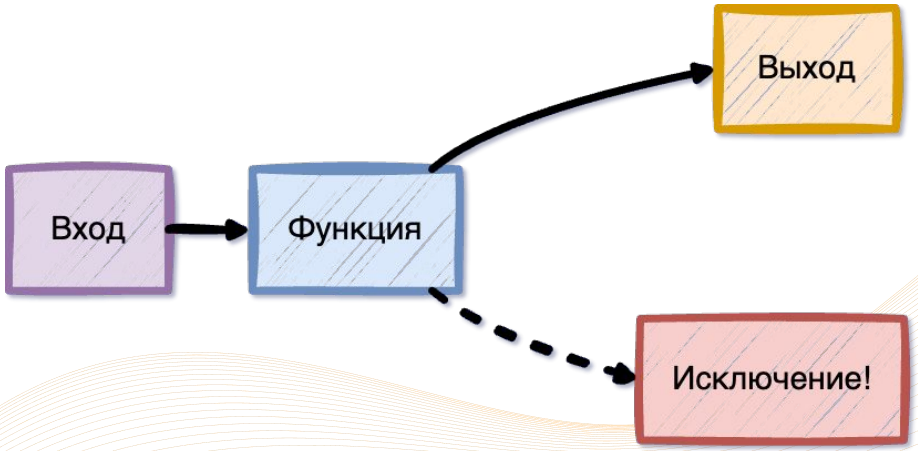
ТРАДИЦИОННЫЕ ЦЕННОСТИ

1. **Скрытый поток управления**: Исключения прерывают выполнение непредсказуемо
2. **Фокус на ошибках**: Код больше занят обработкой ошибок, чем бизнес-логикой
3. **Хрупкость системы**: Легко забыть обработать какой-то случай
4. **Сложность тестирования**: Трудно протестировать все возможные пути выполнения
5. **Потеря контекста**: При исключении теряется информация о промежуточных состояниях

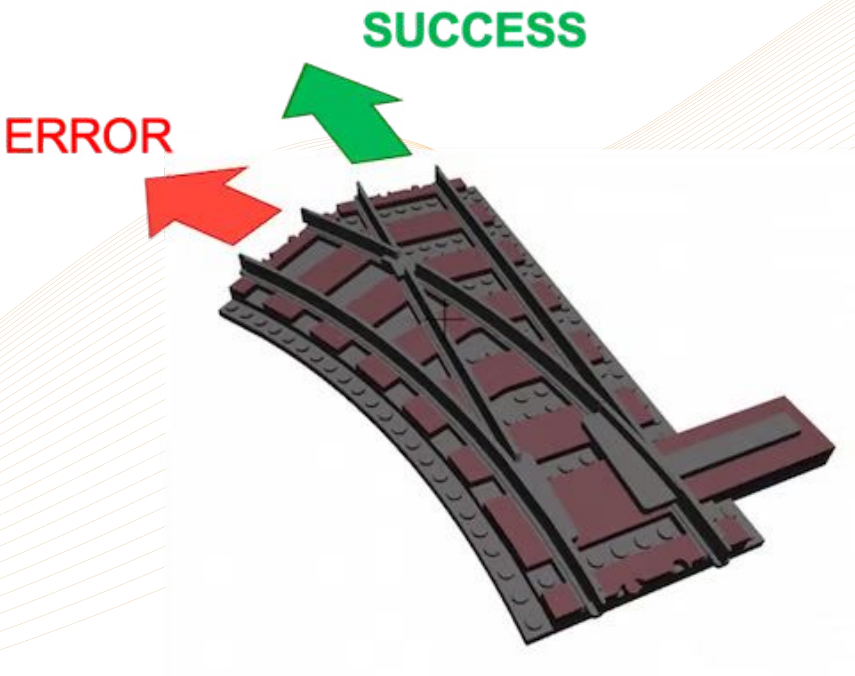
```
exception InvalidUserId
exception EmptyCart
exception ItemsUnavailable of string list
exception PaymentFailed of string

let process_order_bad request =
  if request.user_id ⩽ 0 then raise InvalidUserId;
  let user = find_user request.user_id in
  if user = None then raise (Failure "...");
  if List.length request.items = 0 then raise EmptyCart;
  let unavailable = List.filter (checker) request.items in
  if List.length unavailable > 0 then
    raise (ItemsUnavailable (list unavailable);
  create_order request
```

EVRONE
FP guru

Железная дорога

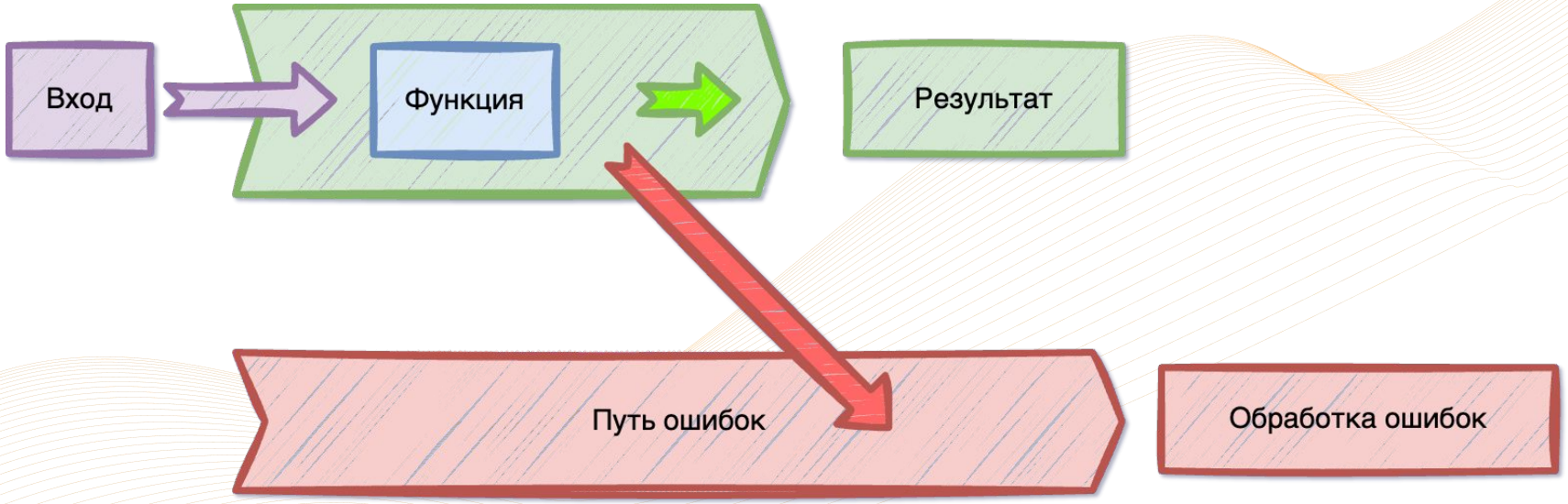


VS



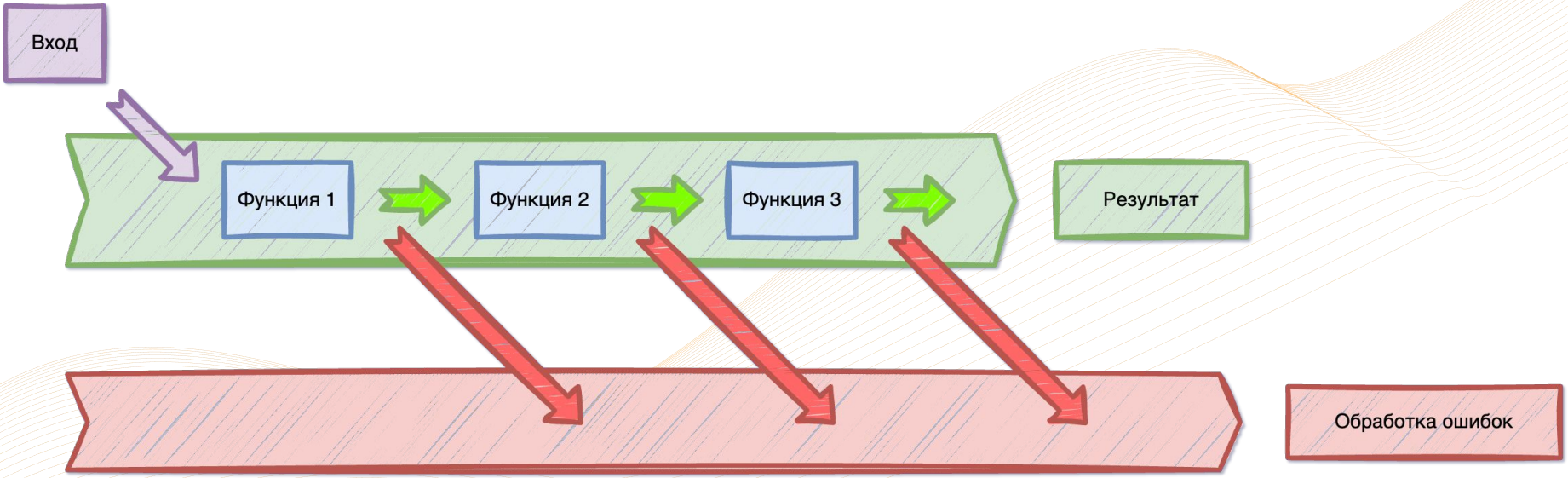
EVRONE
FP guru

Железная дорога



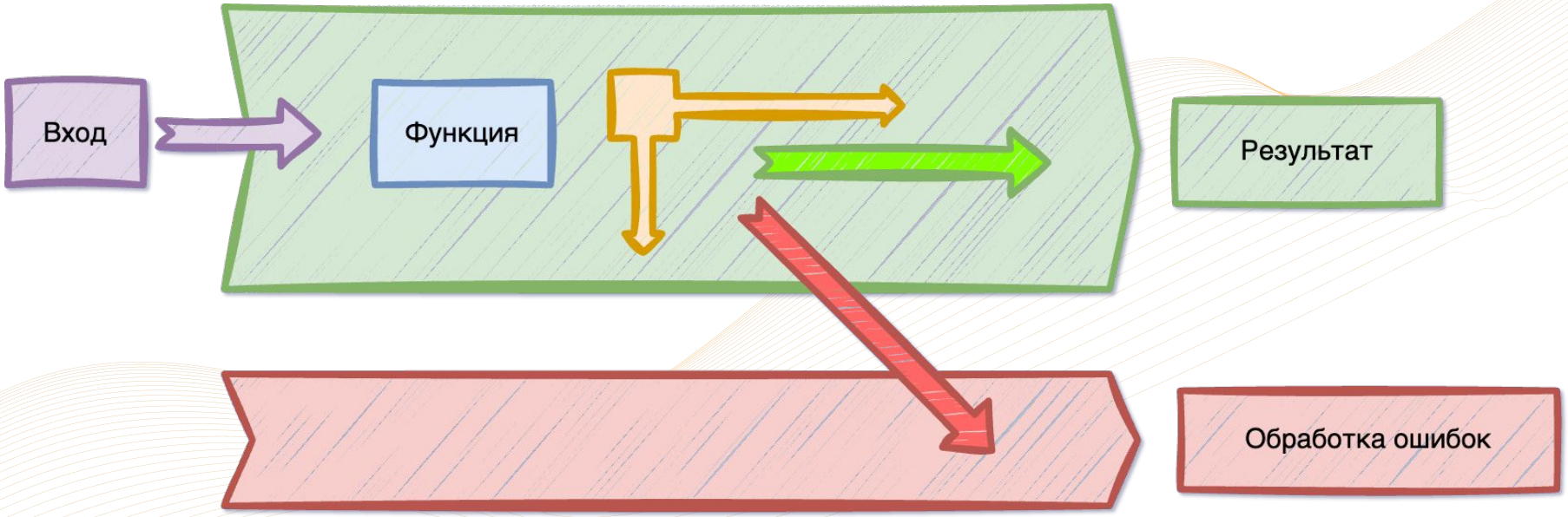
EVRONE
FP guru

Железная дорога



EVRONE
FP guru

Железная дорога



EVRONE
FP guru

Железная дорога: механика

```
type ('success, 'error) result =
  | Ok of 'success                    (* Успешный результат *)
  | Error of 'error                   (* Ошибка *)

type order_error =
  | ValidationError of string         (* Ошибки валидации *)
  | AuthenticationError of int        (* Ошибки аутентификации *)
  | InventoryError of string list     (* Ошибки наличия товаров *)
  | PaymentError of string            (* Ошибки оплаты *)
  | NotificationError of string       (* Ошибки уведомлений *)
```

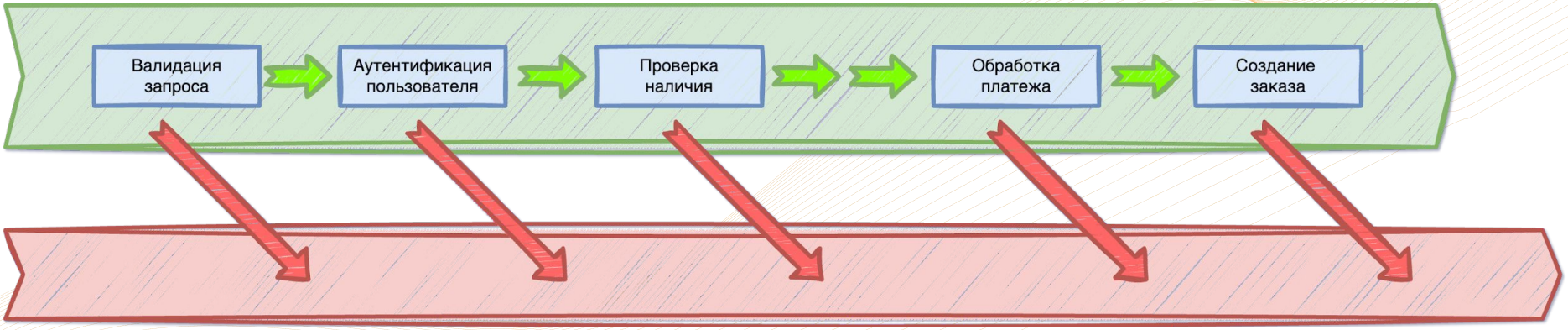
EVRONE
FP guru

Железная дорога: механика

```
(* ← Объявление типов … *)

(* Оператор >>= - железнодорожная стрелка *)
let (>>=) result f =
  match result with
  | Ok value → f value       (* Продолжаем по зеленой колее *)
  | Error e → Error e        (* Остаемся на красной колее *)

(* Оператор >>| - простая езда по зеленой колее *)
let (>>|) result f =
  match result with
  | Ok value → Ok (f value)  (* Применяем функцию к успешному значению *)
  | Error e → Error e        (* Пропускаем ошибку без изменений *)

(* Функция load - обертывание значения в контейнер *)
let load x = Ok x
```

EVRONE
FP guru

Железная дорога: пример



EVRONE
FP guru

Железная дорога: реализация

```
(* ← Механика ЖД *)

(* Основные структуры данных предметной области *)
type cart_item = {
  name: string;                    (* Название товара *)
  price: float;                    (* Цена *)
  quantity: int;                   (* Количество *)
  available: bool;                 (* Доступность *)
}

(* Перечисление способов оплаты *)
type payment_method = CreditCard | PayPal | BankTransfer

(* Структура запроса заказа *)
type order_request = {
  user_id: int;                    (* ID пользователя *)
  items: cart_item list;           (* Список товаров *)
  payment_method: payment_method;  (* Способ оплаты *)
  shipping_address: string;        (* Адрес доставки *)
}
```

EVRONE
FP guru

Железная дорога: реализация

```
(* ← механика ЖД *)
(* ← Доменные типы *)

(* Структура созданного заказа *)
type order = {
  id: string;                                    (* ID заказа *)
  user_id: int;                                  (* ID пользователя *)
  items: cart_item list;                         (* Товары в заказе *)
  total: float;                                  (* Общая стоимость *)
  status: [`Pending | `Confirmed | `Shipped];    (* Статус заказа *)
  created_at: float;                             (* Время создания *)
}
```

EVRONE
FP guru

Железная дорога: реализация

```
(* ← механика ЖД *)
(* ← Доменные типы *)

(* Функция валидации - возвращает Result *)
let validate_request request =
  if request.user_id ⩽ 0 then
    Error (ValidationError "ID пользователя должен быть положительным")
  else if List.length request.items = 0 then
    Error (ValidationError "Корзина не может быть пустой")
  else if String.length (String.trim request.shipping_address) = 0 then
    Error (ValidationError "Адрес доставки обязателен")
  else
    Ok request  (* Валидация прошла успешно *)
```

EVRONE
FP guru

Железная дорога: реализация

```
(* ← механика ЖД *)
(* ← Доменные типы *)
(* ← Валидатор *)

(* Проверка существования пользователя *)
let authenticate_user request =
  let user_exists id = id > 0 && id < 1000000 in
  if user_exists request.user_id then
    Ok request
  else
    Error (AuthenticationError request.user_id)
```

EVRONE
FP guru

Железная дорога: реализация

```
(* ← механика ЖД *)
(* ← Доменные типы *)

(* ← Валидатор *)
(* ← Аутентикатор *)

(* Проверка доступности всех товаров в корзине *)
let check_inventory request =
  let unavailable_items =
    List.filter (fun item → not item.available) request.items
    ▷ List.map (fun item → item.name)
  in
  match unavailable_items with
  | [] → Ok request                          (* Все товары доступны *)
  | items → Error (InventoryError items)  (* Есть недоступные товары *)
```

EVRONE
FP guru

Железная дорога: реализация

```
(* ← механика ЖД *)
(* ← Доменные типы *)

(* ← Валидатор *)
(* ← Аутентикатор *)
(* ← Проверятор *)

(* Обработка платежа с проверкой лимитов *)
let process_payment (request, total) =
  match request.payment_method with
  | CreditCard when total > 50000.0 →
      Error (PaymentError "Превышен лимит кредитной карты")
  | PayPal when total > 25000.0 →
      Error (PaymentError "Превышен лимит транзакции PayPal")
  | _ →
      Ok (request, total, "payment_success")
```

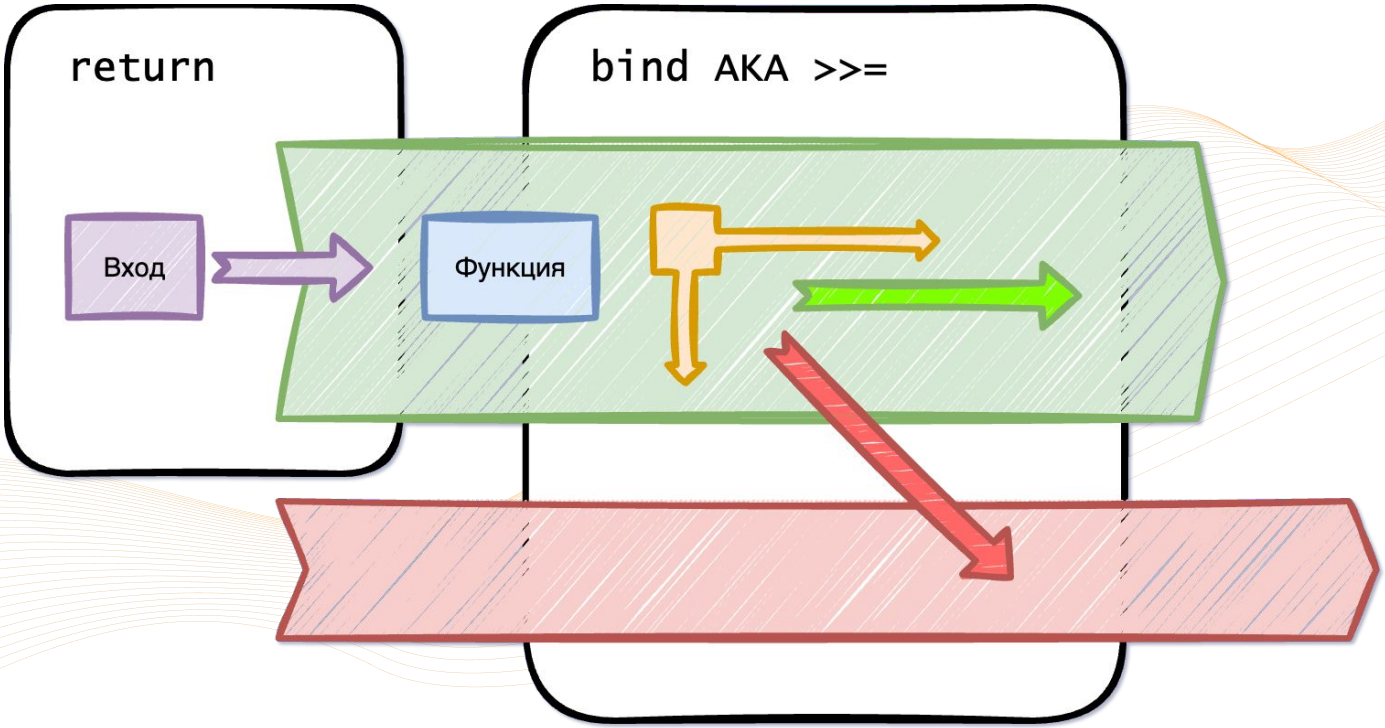
EVRONE
FP guru

Железная дорога: реализация

```
(* ← механика ЖД *)
(* ← Доменные типы *)
(* ← Постадийные функции *)

(* Главная функция обработки заказа *)
let process_order request =
  validate_request request
  >>= authenticate_user
  >>= check_inventory
  >>= calculate_total
  >>= process_payment
  >>= reserve_inventory
  >>= create_order
  >>= send_notifications
```

EVRONE
FP guru

Паровоз: алгебра



EVRONE
FP guru

Паровоз: алгебра

```
(* Тип Result *)
type ('success, 'error) result =
  | Ok of 'success       (* Успешный результат *)
  | Error of 'error      (* Ошибка *)

(* load: помещает обычное значение в монадический контейнер *)
let load x = Ok x
let return = load
and pure = load

(* bind: связывает монадические вычисления *)
let bind m f =
  match m with
  | Ok x → f x
  | Error e → Error e

let (>>=) = bind
```

EVRONE
FP guru

Паровоз: свойства

- **Явные ошибки**: Ошибки становятся частью системы типов
- **Композиция**: Сложная логика строится из простых блоков
- **Безопасность**: Компилятор гарантирует обработку всех случаев
- **Читаемость**: Код фокусируется на бизнес-логике
- **Транзакционность**: Встроенная поддержка атомарных операций
- **Эффективность**: Оптимальное выполнение с минимальными накладными расходами

EVRONE
FP guru

Паровоз: применимость

- Валидация пользовательского ввода
- Интеграция с внешними API
- Парсинг и преобразование данных
- Транзакционные операции
- Построение надежных систем

- Уменьшение количества ошибок в production
- Улучшение читаемости и поддерживаемости кода
- Упрощение тестирования
- Повышение доверия к системе

- Сложные бизнес-процессы с множественными проверками
- Системы, где важна надежность и предсказуемость
- Проекты с требованиями к типобезопасности

EVRONE
FP guru

Паровоз: НЕприменимость

- Простые CRUD операции
- Performance-critical код (хотя накладные расходы минимальны)
- Команды, не готовые к функциональному программированию

EVRONE
FP guru

ЧТО ГУГЛИТЬ?

- Функциональное программирование на <мой_любимый_язык>
- “Монады — не приговор”
- Railway Oriented Programming

ОРЕНБУРГ | 12 — 13 СЕНТЯБРЯ | 2025

EVRONE.RU

FP guru

Павел Аргентов

argentoff@evrone.team

merge