

F#

Примеры функционального решения задач

Московский клуб программистов: <https://www.meetup.com/progmsk/>
Марк Шевченко <https://www.meetup.com/progmsk/members/196915530/>

Пример №1: Обработка исключений

Пример №1: Обработка исключений

```
int count_string(char* s, char* filename)
{
    int count = 0;
    int eof = 0;
    char buffer[BUFFER_SIZE];
    FILE* input;

    if (NULL == s)
        return -1;

    if (NULL == filename)
        return -2;

    input = fopen(filename, "rt");

    if (input == NULL)
        return -3;

    while (NULL != fgets(buffer, BUFFER_SIZE, input)) {
        if (NULL != strstr(buffer, s))
            count++;
    }

    eof = feof(input);
    fclose(input);

    if (eof)
        return count;

    return -4;
}
```

```
int count_string(const string& s, const string& filename)
{
    int count = 0;
    string line;
    ifstream input(filename);

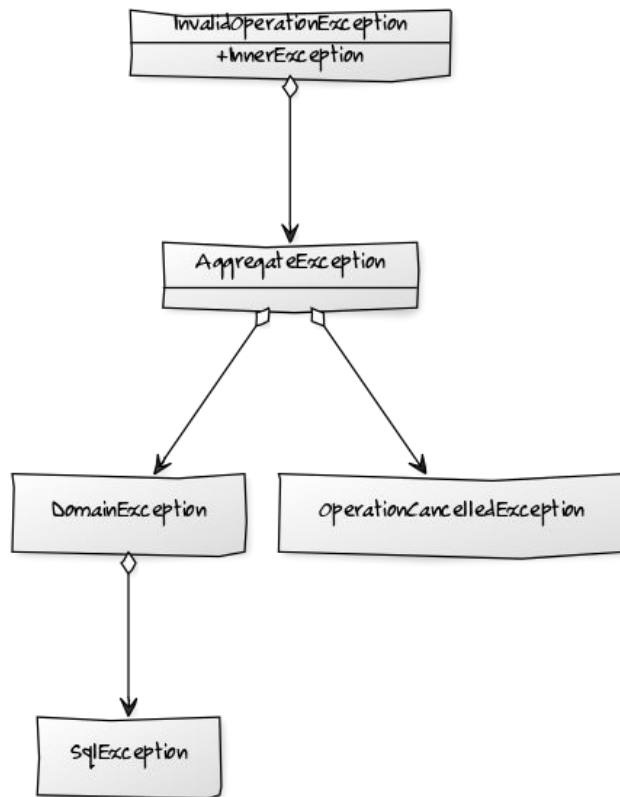
    while (getline(input, line)) {
        if (line.find(s) != string::npos)
            count++;
    }

    return count;
}
```

Пример №1: Внутренние исключения



Пример №1: Внутренние исключения



Пример №1: Throw early, catch later

Пример №1: Throw early, catch later

HTTP/1.1 409 Conflict

```
{
  errors: [
    {
      id: 'email',
      status: 409,
      title: 'Пользователь с таким электронным адресом уже зарегистрирован.',
      detail: 'Кажется, вы уже регистрировались у нас. Если вы забыли пароль, восстановите его.'
    }
  ]
}
```

Пример №1: Throw early, catch later

```
public class UserController : ApiController
{
    private readonly IUserRepository userRepository;
    private readonly DtoMapper dtoMapper;

    ...

    public IHttpActionResult Get(int id)
    {
        try
        {
            var user = userRepository.ReadById(id);
            var userDto = dtoMapper.Map(user);
            return Content(HttpStatusCode.OK, userDto, new JsonMediaTypeFormatter());
        }
        catch (Exception exception)
        {
            var restError = GetRestError(exception);
            return Content(httpError.Status, httpError.Object, new JsonMediaTypeFormatter());
        }
    }
}
```


Пример №1: Глобальный обработчик

```
class JsonApiError
{
    public string Id { get; set; }

    public HttpStatusCode? Status { get; set; }

    public string Title { get; set; }

    public string Detail { get; set; }
}

class JsonApiTopLevelError
{
    public IReadOnlyCollection<JsonApiError> Errors { get; set; }
}

class RestError
{
    public HttpStatusCode Status { get; set; }

    public JsonApiTopLevelError Object { get; set; }
}
```

```
Class RestErrorResult : IHttpActionResult
{
    private readonly HttpRequestMessage requestMessage;
    private readonly RestError restError;

    public RestErrorResult(HttpRequestMessage requestMessage, RestError restError)
    {
        this.requestMessage = requestMessage;
        this.restError = restError;
    }

    public Task<HttpResponseMessage> ExecuteAsync(CancellationToken cancellationToken)
    {
        var response = new HttpResponseMessage(restError.Status);
        var json = JsonConvert.Serialize(restError.Object);
        response.Content = new StringContent(json);
        response.RequestMessage = requestMessage;

        return Task.FromResult(response);
    }
}

class RestExceptionHandler : ExceptionHandler
{
    private readonly IRestExceptionParser parser;

    public RestExceptionHandler(IRestExceptionParser parser)
    {
        this.parser = parser;
    }

    public override void HandleCore(ExceptionHandlerContext context)
    {
        var restError = parser.Parse(context.Exception);
        context.Result = new RestErrorResult(context.ExceptionContext.Request, restError);
    }
}
```

Пример №1: Реализация на C#

```
abstract class Acceptor
{
    public abstract bool Accept(Exception exception);
}

static class ExceptionExtensions
{
    public static void Visit(this Exception exception, Acceptor acceptor)
    {
        if (acceptor.Accept(exception))
            return;

        var aggregateException = exception as AggregateException;
        if (aggregateException != null)
        {
            foreach (var innerException in
                aggregateException.InnerExceptions)
                innerException.Visit(acceptor);
        }

        return;
    }

    if (exception.InnerException != null)
        exception.InnerException.Visit(acceptor);
}
```

```
class RestErrorAcceptor : Acceptor
{
    public HttpStatusCode Status { get; protected set; }

    public string Title { get; private set; }

    RestErrorElement()
    {
        Status = HttpStatusCode.InternalServerError;
        Title = "На сервере возникла внутренняя ошибка.";
    }

    public override bool Accept(Exception exception)
    {
        ...
        return false;
    }
}
```

Пример №1: Паттерн Посетитель

```
public override bool Accept(Exception exception)
{
    var entityException = exception as EntityException;
    if (entityException != null)
    {
        var sqlException = exception.InnerException as SqlException;
        if (sqlException != null)
        {
            if (sqlException.Number == 2601)
            {
                Status = HttpStatusCode.Conflict;

                if (entityException.EntityType == typeof(User))
                    Title = "Пользователь с таким электронным адресом уже существует.";
                else if (entityException.EntityType == typeof(Document))
                    Title = "Документ с таким названием уже существует.";
                else
                    Title = "Сущность с таким названием уже существует.";
            }
        }

        return true;
    }
}
```

```
var invalidOperationException = exception.InnerException as
InvalidOperationException;
if (invalidOperationException != null)
{
    Status = HttpStatusCode.NotFound;

    if (entityException.EntityType == typeof(User))
        Title = "Пользователь с таким идентификатором не найден.";
    else if (entityException.EntityType == typeof(User))
        Title = "Документ с таким идентификатором не найден.";
    else
        Title = "Сущность с таким идентификатором не найдена."
}

return true;
}

return false;
}
```

Пример №1: Реализация на F#

```
let matchWith exceptionType e = if e.GetType() = exceptionType
                                then Some e
                                else None
```

```
let result status message condition e =
    match (condition e) with
    | Some _ -> Some(status, message)
    | None -> None
```

```
let getRestError = result HttpStatusCode.BadRequest
                        "Случилось страшное"
                        (matchWith typedefof<EntityException>)
```

Пример №1: Каррирование

```
let matchWith exceptionType e = if e.GetType() = exceptionType
                                then Some e
                                else None
```

```
let result status message condition e =  
  match (condition e) with  
  | Some _ -> Some(status, message)  
  | None -> None
```

```
let getRestError = result HttpStatusCode.BadRequest
    "Случилось страшное"
    (matchWith typedefof<EntityException>)
```

```
let __hidden_autogenerated_function e = matchWith typedefof<EntityException> e
let getRestError = result HttpStatusCode.BadRequest
                        "Случилось страшное"
                        hidden autogenerated function
```

Пример №1: Операторы

```
let (<|>) rule1 rule2 e =  
    match rule1 e with  
    | Some(_, _) as result -> result  
    | None -> rule2 e
```

```
let getRestError = result HttpStatusCode.BadRequest "Случилось страшное."  
    (matchWith typedefof<EntityException>)
```

```
<|> result HttpStatusCode.BadRequest "Свершилось непоправимое."  
    (matchWith typedefof<InvalidOperationException>)
```

Пример №1: Композиция функций

$$h(g(f(x)))$$

Пример №1: Композиция функций

$$h(g(f(x)))$$

$$h(g(f\ x))$$

Пример №1: Композиция функций

$$h(g(f(x)))$$

$$h(g(f\ x))$$

$$x \mapsto f \mapsto g \mapsto h$$

Пример №1: Композиция функций

```
let getRestError = (matchWith typedefof<EntityException>  
    |> result HttpStatusCode.BadRequest "Случилось страшное.")  
  
    <|> (matchWith typedefof<InvalidOperationException>  
    |> result HttpStatusCode.BadRequest "Свершилось непоправимое.")
```

Пример №1: Предикаты и обобщённые функции

```
let getRestError = matchWith<EntityException>  
    |> where (fun e -> e.EntityType = typeof<User>) |>  
    |> result HttpStatusCode.BadRequest "Случилось страшное."
```

Пример №1: Предикаты и обобщённые функции

```
let getRestError = matchWith<EntityException>  
    |> where (fun e -> e.EntityType = typedefof<User>)  
    |> result HttpStatusCode.BadRequest "Случилось страшное."
```

```
let matchWith<'E when 'E :> Exception> (e: Exception) =  
    if e :? 'E  
    then Some (e :?> 'E)  
    else None
```

```
let where<'E when 'E :> Exception> predicate condition (e: Exception) =  
    match (condition e) with  
    | Some e -> if (predicate e)  
                  then Some e  
                  else None  
    | None -> None
```

Пример №1: Внутренние исключения

```
let getRestError = (matchWith<EntityException>  
    |> where (fun e -> e.EntityType = typeofof<User>)  
    |> inner<EntityException, InvalidOperationException>  
    |> result HttpStatusCode.BadRequest "Случилось страшное.")
```

Пример №1: Внутренние исключения

```
let getRestError = (matchWith<EntityException>  
    |> where (fun e -> e.EntityType = typedefof<User>)  
    |> inner<EntityException, InvalidOperationException>  
    |> result HttpStatusCode.BadRequest "Случилось страшное.")
```

```
let inner<'E1, 'E2 when 'E1 :> exn and 'E2 :> exn> (condition: 'E1 -> 'E1 option) (e: 'E1) =  
    let tryFindInAggregate (ae: AggregateException) =  
        let found = Seq.tryFind (fun ie -> box ie :? 'E2) ae.InnerExceptions  
        match found with  
        | Some ie -> Some (ie :?> 'E2)  
        | None -> None
```

```
    let tryFindInInner (e: exn) =  
        let ie = e.InnerException  
        match ie with  
        | :? 'E2 -> Some (ie :?> 'E2)  
        | _ -> None
```

```
    match (condition e) with  
    | Some e' -> if (box e' :? AggregateException)  
        then tryFindInAggregate (box e' :?> AggregateException)  
        else tryFindInInner e'  
    | None -> None
```

Пример №1: Свод правил

```
let getRestError = (matchWith<EntityException>
    |> where (fun e -> e.EntityType = typeofof<User>)
    |> inner<EntityException, InvalidOperationException>
    |> result HttpStatusCode.BadRequest "Случилось страшное.")

<|> (matchWith<EntityException>
    |> where (fun e -> e.EntityType = typeofof<Document>)
    |> inner<EntityException, SqlException>
    |> where (fun e -> e.Number = 2601)
    |> result HttpStatusCode.Conflict
        "Пользователь с таким электронным адресом уже существует.")
```

Пример №1: Выводы

1. Код понятен даже при беглом изучении.

Пример №1: Выводы

1. Код понятен даже при беглом изучении.
2. Код проще аналогичного императивного кода.

Пример №1: Выводы

1. Код понятен даже при беглом изучении.
2. Код проще аналогичного императивного кода.
3. Обслуживающий код тоже проще.

Пример №1: Выводы

1. Код понятен даже при беглом изучении.
2. Код проще аналогичного императивного кода.
3. Обслуживающий код тоже проще.

Сопровождающий текст ищите по адресу

<https://github.com/markshevchenko/lectures/tree/master/f-sharp/exception-handling>