

Тема: Разработка программного модуля для автоматизированного тестирования веб-форм (Шифр ПМ АТВ)

Руководитель от СПИНТех: к.п.н., доц. Федотова Елена Леонидовна

Исполнитель: ст. гр. ПИН-51Д Джугели Дмитрий Александрович

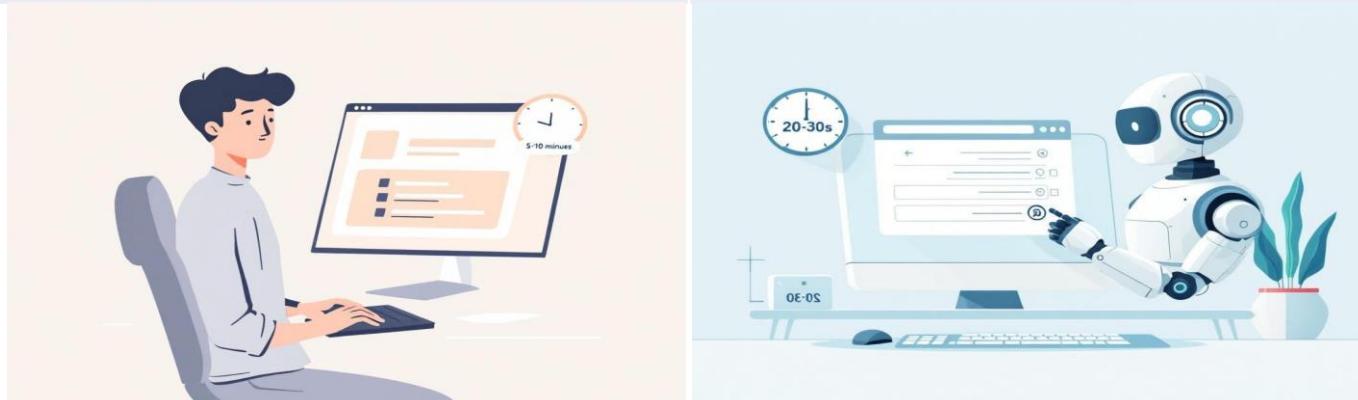
Цель: сокращение временных затрат на тестирование веб-форм с email-верификацией.

Задачи:

- исследование предметной области;
- сравнительный анализ существующих аналогов;
- выбор языка и среды программирования;
- разработка концептуальной модели;
- разработка схемы данных и API;
- разработка пользовательского интерфейса;
- программная реализация модуля ПМ АТВ;
- тестирование модуля ПМ АТВ;
- разработка руководства оператора.

Исследование предметной области

До разработки ПМ АТВ	После разработки ПМ АТВ
1. Ручное создание тестовых данных и почтовых ящиков занимает 5-10 минут на сценарий	1. Полная автоматизация процесса занимает 20-30 секунд на сценарий
2. Риск ошибок из-за ручного ввода и повторного использования email	2. Автоматическая генерация уникальных данных исключает человеческие ошибки
3. Постоянное переключение между браузером и почтовым клиентом	3. Единая среда работы в браузере с просмотром писем в реальном времени
4. Требуются навыки программирования для автоматизации процессов	4. Доступность инструмента для специалистов без глубоких технических знаний



Обзор существующих программных решений

Критерий	Программные решения				
	Guerrilla Mail ^[1]	Temp-Mail ^[2]	Selenium ^[3] + Faker ^[4]	MailTrap ^[5]	ПМ АТВ
Интеграция в браузер (расширение)	-	-	+-	-	+
Генерация данных (российский регион)	-	-	+	-	+
Автоподстановка в формы	-	-	+	-	+
Просмотр писем в реальном времени	+-	+-	-	+	+
Наличие временной почты	+	+	-	+	+
Стоимость подписки (руб/мес)	785	800	0	1200	0
Локализация (интерфейс на русском языке)	+	-	+-	-	+

Источники информации:

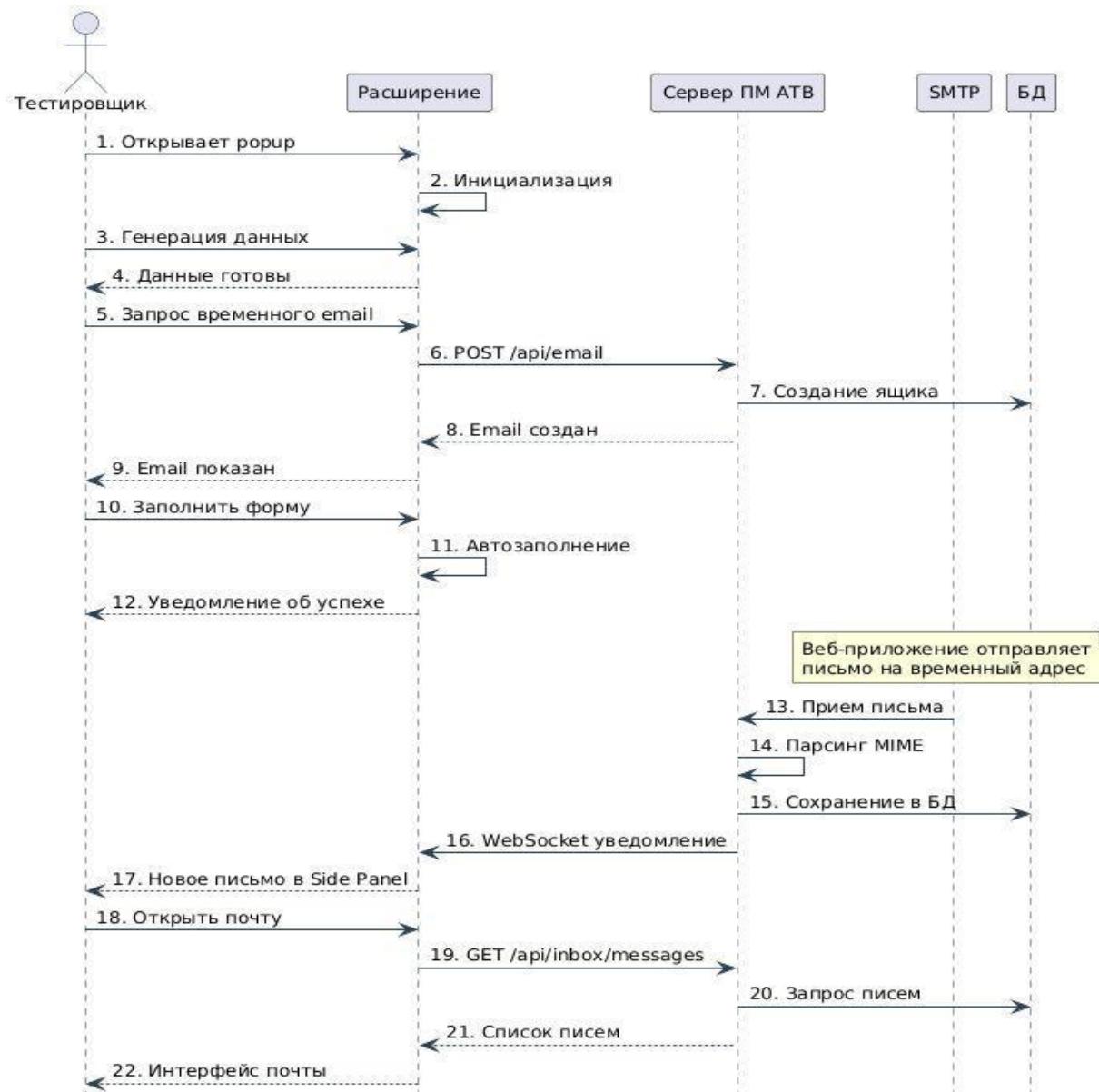
- [1] <https://www.guerrillamail.com/>
- [2] <https://temp-mail.org/>
- [3] <https://www.selenium.dev/>
- [4] <https://fakerjs.dev/>
- [5] <https://mailtrap.io/>

Условные обозначения:

- «+» - присутствует
- «+-» - присутствует частично
- «-» - отсутствует

Фрагмент концептуальной модели. Диаграмма последовательности

4



Выбор языка программирования для серверной части

Критерий сравнения	Язык				
	Java ¹	C# ²	Go ³	Node.js ⁴	Python ⁵
Производительность, запросы в секунду	~2500	~3000	~3500	~2000	~2500
Поддержка асинхронности	+ CompletableFuture	+ async/await	+ goroutine	+ event loop	+ asyncio
Библиотеки для SMTP/Email	+ JavaMail	+ MailKit	+ Stdlib	+ Nodemailer	+ Aiosmtpd
Поддержка WebSocket	+ Spring WS	+ SignalR	+ Gorilla WS	+ Socket.IO	+ Websockets
Популярность языка (место в рейтинге IEEE Spectrum)	2	7	8	Нет данных	1
Опыт разработки (лет)	0,5	1	0	0	1

Источники информации:

- [1] <https://www.java.com>
- [2] <https://dotnet.microsoft.com/languages/csharp>
- [3] <https://go.dev/>
- [4] <https://nodejs.org/>
- [5] <https://www.python.org/>
- [6] <https://spectrum.ieee.org/top-programming-languages-2024>

Условные обозначения:

- «+» - присутствует
- «+-» - присутствует частично
- «-» - отсутствует

Выбор языка программирования для клиентской части

Критерий сравнения	Язык			
	TypeScript ¹	Dart ²	WebAssembly ³	JavaScript ⁴
Совместимость с браузерами	Компилируется в JS	Требует компиляции	Требует JS-прослойки	Нативный
Прямой доступ к DOM и API браузера	Полный	Ограниченный (через js-interop)	Отсутствует (только через JS)	Полный
Популярность языка (место в рейтинге IEEE Spectrum[5])	5	46	19	3
Время компиляции/сборки (сек)	1-3	3-10	30+	0
Размер runtime (КБ)	0	~50 КБ	~100 КБ	0
Опыт разработки (лет)	0,25	0	0	0,5

Источники информации:

[1] <https://www.typescriptlang.org/>

[2] <https://dart.dev/>

[3] <https://webassembly.org/>

[4] <https://www.javascript.com/>

[5] <https://spectrum.ieee.org/top-programming-languages-2024>

Выбор среды разработки

Параметры сравнения	Среда разработки			
	PyCharm Pro ¹	WebStorm ²	IntelliJ IDEA ³ Ultimate	Visual Studio ⁴ Code
Поддержка Python	+	-	+	+
Поддержка JavaScript	+/-	+	+	+
Интеграция с Git	+	+	+	+
Количество доступных плагинов, штук	~3000	~2000	~5000	~40000
Производительность, секунд на запуск	~5-10	~5-10	+~10-15	~1-5
Стоимость, рублей в год	~5000	~4500	~6000	Бесплатно

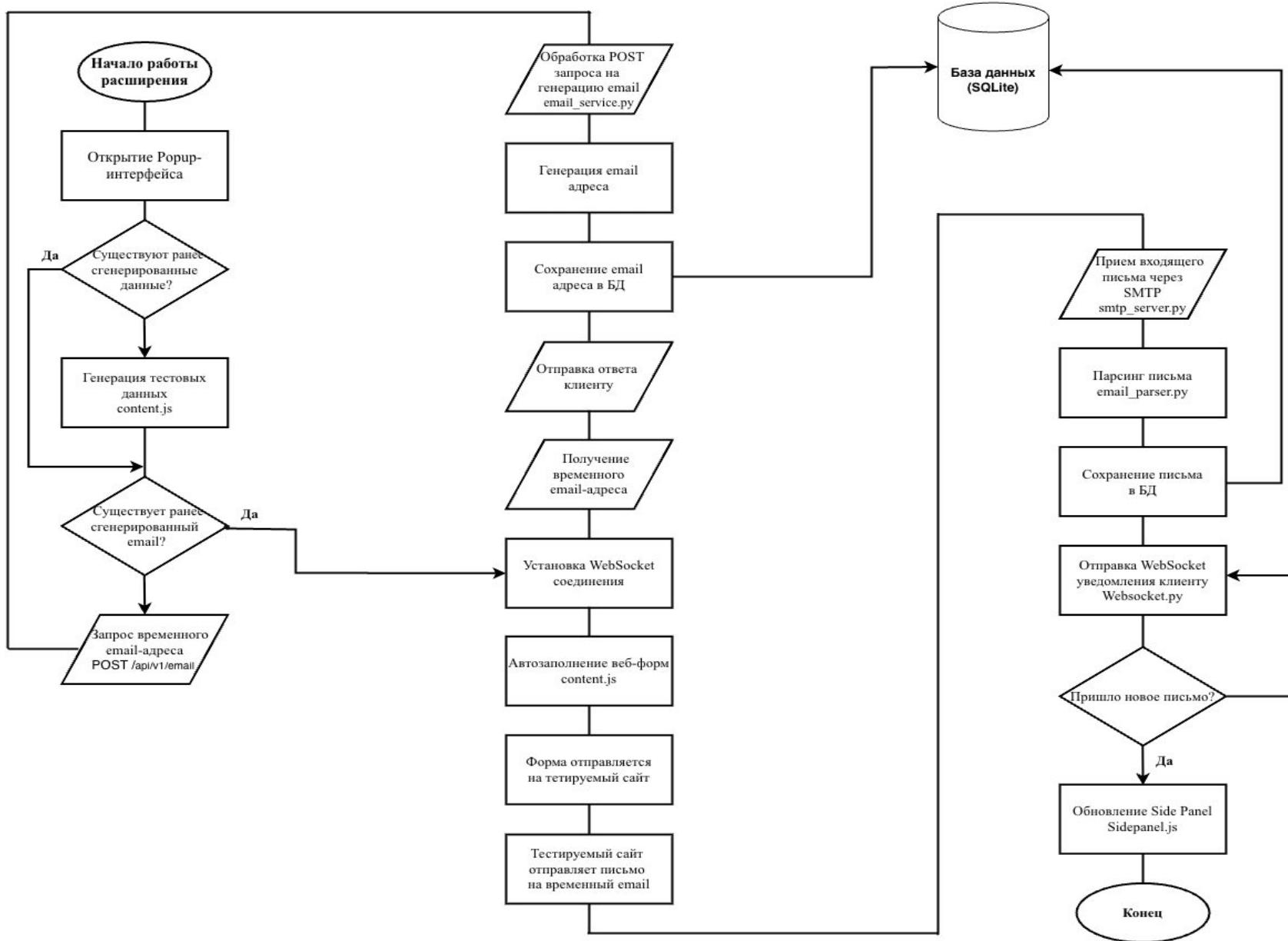
Источники информации:

- [1] <https://www.jetbrains.com/pycharm/>
- [2] <https://www.jetbrains.com/webstorm/>
- [3] <https://www.jetbrains.com/idea/>
- [4] <https://code.visualstudio.com/>

Условные обозначения:

- «+» - присутствует
- «+/-» - присутствует частично
- «-» - отсутствует

Схема алгоритма ПМ АТВ



Пользовательский интерфейс. Экранная форма расширения

Временный Email

Создать email

n506qgqlnv@temp.atv.local

Отключено

Тестовые данные

Сгенерировать данные

Имя: Екатерина
Фамилия: Макаров
ФИО: Павлов Роман Алексеевна
Телефон: +7 (495) 6502-4222
Email: n506qgqlnv@temp.atv.local
Адрес: г. Воронеж, ул. Новая, д. 167, кв. 81
Компания: ООО "Дизайн Студия"

Действия

Заполнить форму на странице

Очистить данные

Рис.1 Основная экранная форма расширения

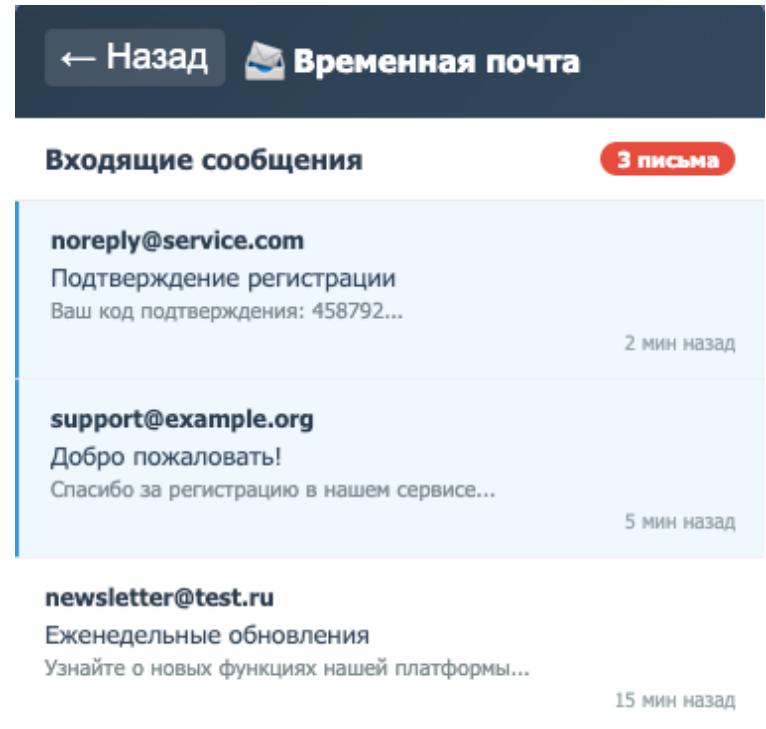


Рис.2 Экранная форма входящих сообщений

Тестирование ПМ АТВ

File ▲	statements	missing	excluded	coverage
app/__init__.py	3	0	0	100%
app/api/__init__.py	4	0	0	100%
app/api/v1/__init__.py	2	0	0	100%
app/api/v1/endpoints.py	52	15	0	70%
app/api/v1/websocket.py	60	12	0	80%
app/core/__init__.py	2	0	0	100%
app/core/config.py	11	0	0	100%
app/database.py	13	4	0	69%
app/main.py	29	2	0	93%
app/models.py	22	3	0	86%
app/schemas.py	28	7	0	75%
app/services/__init__.py	2	0	0	100%
app/services/email_parser.py	47	7	0	85%
app/services/email_service.py	11	1	0	91%
app/services/smtp_server.py	78	13	0	83%
Total	364	64	0	82%

coverage.py v7.13.1, created at 2026-01-15 19:10 +0300

Рис.3 Отчет о покрытии кода

```
Запуск интеграционных тестов ПМ АТВ
=====
Health check passed: {'status': 'healthy'}
Root endpoint: ПМ АТВ API работает! 🚀

Начало интеграционного теста...
1. Создаю временный email...
Email создан: 7ffus5c9x3@temp.atv.local
ID: 465ed6e2-5a51-420d-9054-93a3d400535c
2. Получаю информацию о email...
Информация получена
3. Проверяю входящие письма...
Входящих писем: 0
4. Ищу email по адресу...
Email найден по адресу
5. Удаляю тестовый email...
Email удален

Рабочий цикл завершен: {'success': True, 'email_created': '7ffus5c9x3@temp.atv.local', 'steps_completed': 5}

✖ Тестирую обработку ошибок...
1. Запрос несуществующего email...
Несуществующий email: 404 OK
2. Запрос несуществующего эндпоинта...
Несуществующий эндпоинт: 404 OK
3. Запрос с некорректным UUID...
Некорректный UUID: 422 OK
Обработка ошибок: {'success': True, 'errors_tested': 3}
=====
```

Рис.4 Результаты интеграционного тестирования

```
ant_daddy@Dmitriy backend % pytest tests/ -v
collected 9 items

tests/test_email_service.py::test_email_generation PASSED
[ 11%]
tests/test_email_service.py::test_email_format PASSED
[ 22%]
tests/test_email_service.py::test_expiry_calculation PASSED
[ 33%]
tests/test_example.py::test_addition PASSED
[ 44%]
tests/test_example.py::test_email_generation PASSED
[ 55%]
tests/test_example.py::test_parse_email PASSED
[ 66%]
tests/test_module.py::test_basic_addition PASSED
[ 77%]
tests/test_module.py::test_string_operations PASSED
[ 88%]
tests/test_module.py::test_list_operations PASSED
[100%]
```

Рис.4 Выполнение модульного тестирования

Достижение цели ВКР

Цель: сокращение временных трудозатрат на процесс тестирования веб-форм с email-верификацией

Доказательство:

Критерий оценки эффективности	Ручное тестирование (приблизительно)	С использованием ПМ АТВ (приблизительно)
Генерация тестовых данных (ФИО, адрес, телефон)	1-2 мин. (поиск, копирование)	<0.5 сек (автоматически)
Заполнение полей формы	2-4 мин. (ручной ввод)	1-2 сек (автоподстановка)
Создание временного email	2-5 мин. (регистрация на стороннем сервисе)	<0.5 сек (автоматически)
Ожидание и получение письма	2-5 мин. (переключение между вкладками браузера, проверка почты)	<30 сек Автоматическое отображение в интерфейсе расширения
Полный цикл тестирования одного сценария	5-10 мин.	1 мин.

Вывод: разработанный программный модуль позволяет автоматизировать ключевые этапы тестирования веб-форм, что приводит к сокращению временных затрат и повышению надежности процесса тестирования.

Результаты работы

- исследована предметная область;
- проведен сравнительный анализ существующих аналогов;
- выбраны язык и среда разработки;
- разработана концептуальная модель ПМ АТВ;
- разработан алгоритм работы ПМ АТВ;
- разработана схема данных и API;
- разработан пользовательский интерфейс;
- осуществлена программная реализация модуля ПМ АТВ;
- проведено тестирование модуля ПМ АТВ;
- разработано руководства оператора для ПМ АТВ.

Спасибо за внимание!