

# FastRxJava

이승민

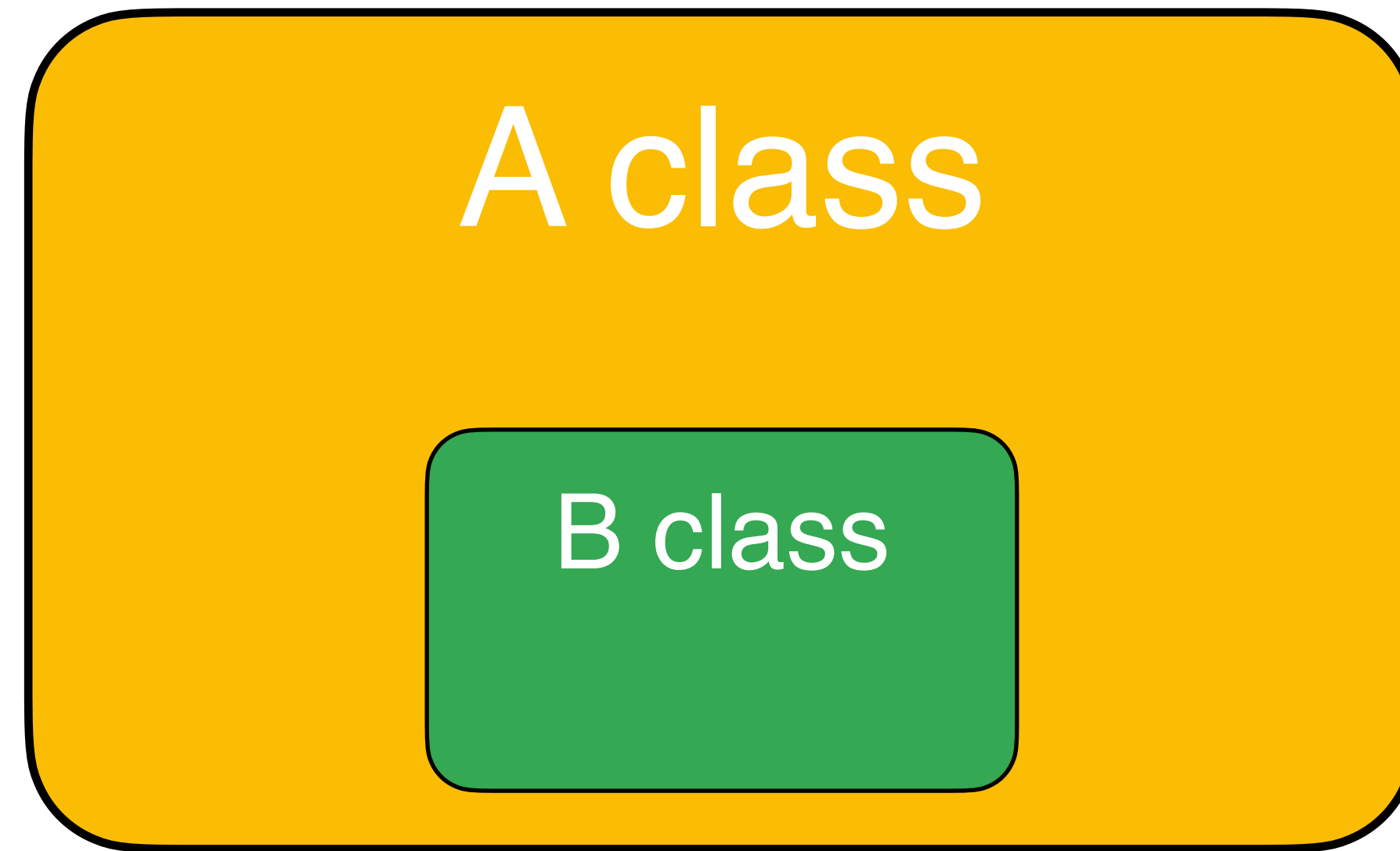
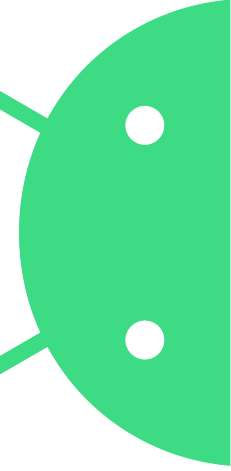
뱅크샐러드 Android Developer / GDE Android Korea

[rfrost77@gmail.com](mailto:rfrost77@gmail.com)



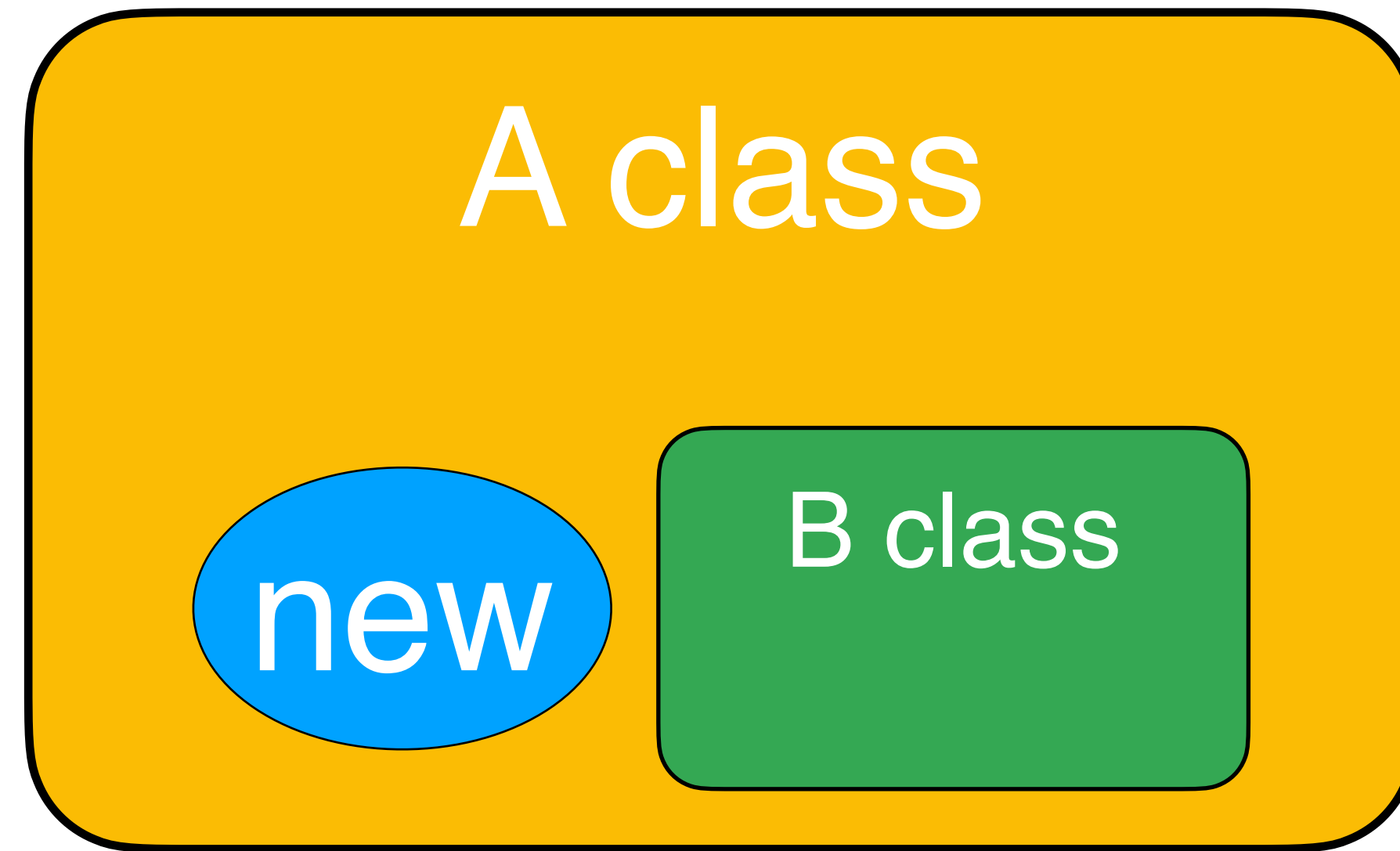
# 메모리 관리

# Dependency - 의존성?

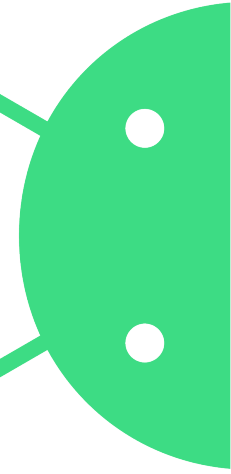


A는 B에 의존

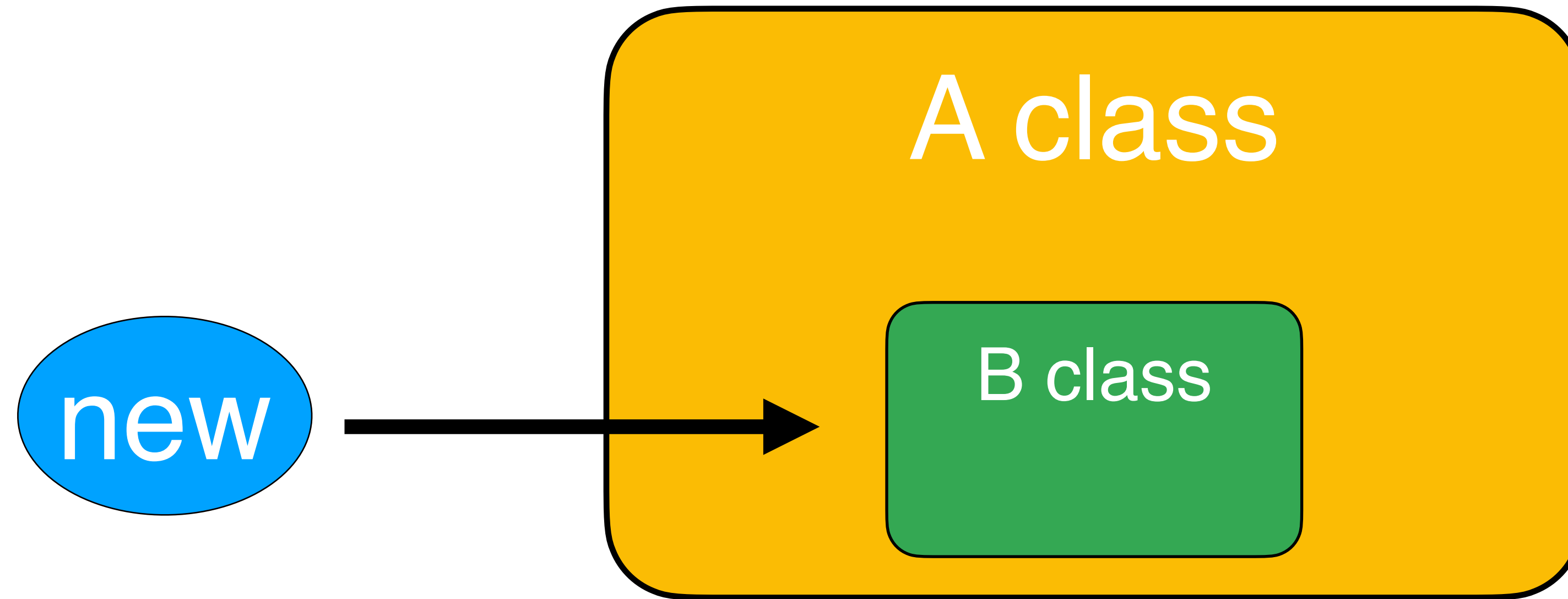
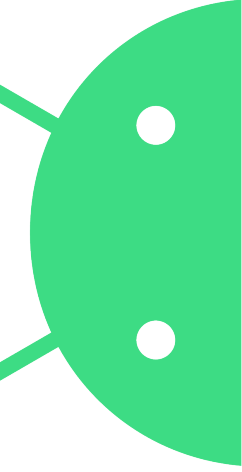
# 의존성 생성?



new로 생성



# 의존성 주입?



생성된 의존성 객체를 전달받음

```
class Car {  
    private val engine: Engine  
}
```



```
class Car {  
    private val engine: Engine  
}
```

# 의존성



```
class Car {  
    private val engine: Engine = Engine()  
}
```





```
class Car {  
    private val engine: Engine = Engine()  
}
```

# 의존성 생성



```
class Car(  
    private val engine: Engine  
)
```

**Constructor  
Parameter**

의존성 주입



# 가비지 컬렉터

## 의존성이 끊긴 객체를 회수

## 메모리 재할용

# 메모리 누수

사용 안하지만 의존성이 안끊겨서 회수되지 않는 객체

메모리가 재활용되지 않음

# 안드로이드의 메모리 누수

**Activity는 사라졌지만 Context가 살아있음**

**Activity 객체가 사라지지 않음**

# Context

**Activity/Fragment 의 정보를 담고있음**

**Theme/BackStack 등**

# RxJava의 메모리 누수

Activity는 사라졌지만

Context를 담은 Observer가 살아있음

# 어떻게 메모리 누수 해결?

수동으로 구독을 끊는다

`Disposable.dispose()`



# Disposable

# Observer

## `subscribe()` / `subscribeWith()` 의 반환값

# Disposable

**subscribe(Consumer)**

**subscribeWith(Observer)**

# CompositeDisposable

Disposable 모음

onDestroy()에서 dispose()

**onComplete(), onError() 시 자동 dispose()**

**대부분 자동 dispose()**

# **dispose() 필요 케이스**

**Observer가 View 동작을 포함**

**Observer 동작 시 View가 사라져있으면 에러**

# **dispose() 필요 케이스**

**Single, Completable이 아닌**

**지속적인 Observable, Flowable**



# 구독 원리 2

**onComplete(), onError() 시 자동 dispose()**

**대부분 자동 dispose()**



**fromCallable()**

**Disposable 처리가 된 create()**

# Exception



**Rx는 수행 중 에러가 나면 onError()  
이후 스트림은 동작하지 않는다**

**Rx는 수행 중 에러가 나면 onError()**

**이후 스트림은 동작하지 않는다**

**Error 받기**

**Consumer**

**Observer.onError()**

# Error 받기

## Observer.onError()

## Observer를 상속받아 활용하자

# Error 받기

Observer를 상속받아 활용하자

Default Error 처리

ex) 로그, 전역 에러

# 로그

## e.printStackTrace()

### Analytics, CrashReport 등 서드파티



# 전역에러

## Event로 처리

### Toast, Dialog, Activity 등 보여주기

# 에러처리 어디에서?

**View**

**ViewModel**

**UseCase**

**Repository**

**HTTP**

# 에러처리 어디에서?

**View**

**ViewModel**

**UseCase**

**Repository**

**HTTP**

**어디든 담당로직에 맞게**

# One Stream 관리

OnError()가 Stream 단위

Stream이 끊어지면 onError()를 받지 못한다

subscribe(subscribe()) vs flatMap()

Rx는 수행 중 에러가 나면 onError()

**이후 스트림은 동작하지 않는다**

# Operators

**onErrorResumeNext**

**onErrorComplete**

**onErrorReturn**

# Default Error 처리

## RxJavaPlugin

끝

이승민

뱅크샐러드 Android Developer / GDE Android Korea

[rfrost77@gmail.com](mailto:rfrost77@gmail.com)