VIEWMODEL

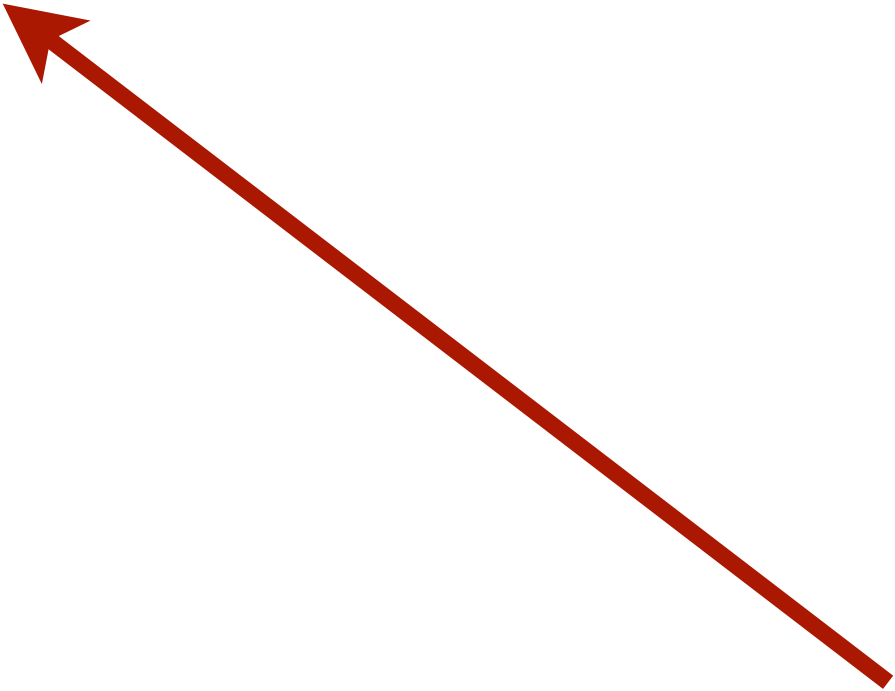
```
abstract class ViewModel {
    protected fun onCleared() {}
}
```

28

```
abstract class BaseViewModel : ViewModel() {

    open fun close() {
        //clear resources
    }

    override fun onCleared() {
        close()
        super.onCleared()
    }
}
```



invoke close in test flow

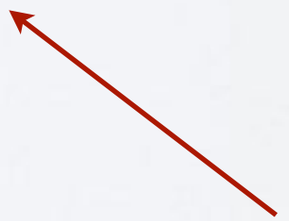
VIEWMODEL

```
abstract class ViewModel {
    protected fun onCleared() {}
}

abstract class BaseViewModel : ViewModel() {

    open fun close() {
        //clear resources
    }

    override fun onCleared() {
        close()
        super.onCleared()
    }
}
```



invoke close in test flow

MODEL-VIEW-VIEWMODEL

```
class SignUpViewModel(
        private val useCase: SignUpUseCase,
        private val router: Router
) : BaseViewModel() {

    var progressVisible by observable(false)

    fun signUpClick() {
        useCase.execute(signUpEntity)
          .doOnSubscribe { progressVisible = true }
          .doOnComplete { progressVisible = false }
          .subscribe({},{})
    }

    fun signInClick() {
        router.openSignIn()
    }
}
```