FACTORY METHODS + DEFAULT ARGUMENTS

```
fun createPresenter(
        router: Router = this.router,
        signUpView: SignUpView = this.signUpView,
        userProvider: UserProvider = this.userProvider,
        formValidator: FormValidator = this.formValidator,
        signUpUseCase: SignUpUseCase = this.signUpUseCase
): SignUpPresenter {
    return SignUpPresenter(
            signUpView,
            router,
            formValidator,
            userProvider,
            signUpUseCase
    )
}
```

FACTORY METHODS + DEFAULT ARGUMENTS

```
@Test
fun `given … when … then propagate error to view`(){
    val useCase: SignUpUseCase = mock {
        on { execute(any(), any()) } doThrow Exception()
    }
    val presenter = createPresenter(
            signUpUseCase = useCase
    )

    verify(signUpView).showError()
}
```