REPOSITORY = 'https://github.com/betrakiss/ruby-retrospective-2015-1'

# Неща, които научих от ретроспекцията

# 0. Hash-овете са хубав заместител вместо дълъг case

# 1. Да извеждам каквото мога отвън, като клас променлива

# 2. Да пиша имената на функциите си по-свързани

# с бизнес стойността им към проекта

# 3. На места, вместо по кратък и неразбираем код,

# да пиша по-дълъг, но разбираем (ако изключим алгоритми..)

# 4. Хубавото индентиране има значение.

# 5. Правенето на wrapper-и на някои готови функции носи

# по-голяма разбираемост на кода в текущия контекст

# 6. Да слагам private по-често

# 7. Lazy е хубаво нещо

# 8. Вградените методи винаги имат и по още някой аргумент,

# който помага за екстра случаи (reduce(1, :+))

# 9. var || 1 е хубав начин да сложиш 1 по default

# (all hail javascript..)

# 10. Проверката с gcd e много добра,

# за да може всяко рационално число да присъства само веднъж

# 11. За проверка дали едно число е просто,

# може да се проверят числата само до корен от целта,

# а не до самата цел

# 12. enum\_for прави each-a много по изчистен и удобен

# 13. Partition много лесно разцепва нещо, когато има нужда

# 14. Struct.new спестява създаването на accessor-и

# и е удобно за малки класове

# 15. Скобите са излишни, когато функцията не приема параметри

# 16. Скобите не са излишни когато има един параметър (това не е perl.. за щастие :D)

# 17. Имената с много долни черти са неприятни за окото

# 18. Ако един клас е нужен само и единствено на точно един друг, няма смисъл класът да не е вложен във втория

# 19. Ако имам вложени класове е по-ясно, ако те са най-отгоре на бащиния клас

# 20. Функционалното програмиране доста скъсява кода

# и е много по-лесно четимо) (map, reduce,select..)

# 21. all? и any? могат да спестят много логика и редове код

# 22. Научих и нова дума – consecutive ^^