# 🐈

## Структура проекта

Предлагается использовать следующую структуру проекта для всех лабораторных работ сразу:

1. src/ (Top directory)
2. lab1/ (Package)
3. Rational.java (Class)
4. ...
5. Main.java (Runnable)
6. lab2/ (Package)
7. Artist.java (Class)
8. ...
9. Main.java (Runnable)
10. playground/ (Package)
11. TryingStreams.java (Runnable)
12. ISummable.java (Interface)
13. SummableImplementation.java (Class)
14. SomethingMore.java (Runnable)
15. IDoNotKnowWhatIsThat.java (Runnable)
16. Utils.java (static-Class)
17. tests/ (Package)
18. lab1/ (sub-Package)
19. TestRational.java (unit-test)
20. TestRationalList.java (unit-test)
21. TestPolynom.java (unit-test)
22. TestIntegration.java (intergration test)
23. lab2/ (sub-Package)
24. TestArtist.java (unit-test)
25. ...
26. TestIntegration.java (intergration test)

* Если вы используете Maven в IDEA, то, скорее всего, ваш проект будет автоматически структурирован немного иначе (src/{main,test}/java/lab{1,2,…}).
* Тесты необязательны, но желательны!
* Запускаемый файл Main.java содержит в себе код, проверяющий работоспособность всех реализованных классов и их взаимодействий. Хорошо написанный Main.java может заменить собой тесты, если покрыты все краевые случаи и сценарии использования классов.
* Пакет playground включает в себя всё, что вам угодно, не относится к лабам, и существует лишь затем, чтобы не создавать отдельный проект для всяких мелочей, которые вы хотите попробовать.

## 

## Во время написание кода

Пожалуйста, пишите красивый код, а именно:

* Имя класса/интерфейса – существительное в единственном числе, с большой буквы в стиле TitleCase.
  + RationalList, ClientFactory, EntityBuilder, Task, CalculationStrategy
* Имя функции (метода) – глагол, отвечающий за то, что делает функция, с маленькой буквы в стиле camelCase.
  + getName, isAvailable, countLargerThan, getOptimalRoute, runApp
* Имя локальной переменной – ёмкое существительное, нуВыПонялиВКакомСтиле.
  + inputFileName, taskStatus, idToNameMapping (id2name/mapIdName)
* Имя поля класса похоже на локалку, но не должно содержать имени класса:
  + name, но не planetName
  + isDone, но не isTaskDone
  + position, но не objectPosition
* Старайтесь избегать однобуквенных имён, кроме:
  + всеми любимых переменных в цикле (i,j,k)
  + временных переменных с понятным предназначением (String s, Time t)
  + координат (x,y,z)

Пожалуйста, старайтесь реализовать в своих классах следующие методы:

* String toString()
* …

Пожалуйста, используйте [IDEA](https://www.jetbrains.com/idea/download).

## 

## Перед сдачей лабы

* Ctrl+Alt+L позволит вам автоматически форматировать код в соответствии с принятым в джаве code-style’ом. Также можно включить реформат импортов (в Ctrl+Shift+Alt+L), что позволит автоматически приводить импорты в опрятный вид. Форматирование можно тонко настроить там же, где и всегда: Ctrl+Alt+S.
* Посмотрите на все подсвечиваемые IDEA’ей участки кода (ошибки/варнинги) и постарайтесь их исправить с помощью Alt+Enter. Однако, в некоторых случаях следует делать это аккуратно – не всегда стоит соглашаться на, например, замену “public” на “package-private”.
* Запустите ваш код (Main.java) хотя бы раз и проверьте, что выводимые в консоли сообщения репрезентативно отражают работу и взаимодействие написанных вами классов.
* Удостоверьтесь, что вы проверили все, или хотя бы основные, крайние и граничные случаи:
  + отрицательные числа
  + нецелые числа
  + другие странные числа
  + пустые строки
  + большие строки
  + кривые строки
  + неправильный порядок инициализации объектов
  + и многое другое