

# ФИТ НГУ, курс Haskell, осенний семестр 2019

## Квантовые вычисления

### Общие сведения

В далеком будущем, которое мы с Вами застанем, весьма актуальны будут программисты на квантовых компьютерах. По счастливому совпадению, квантовые вычисления отлично описываются на языке Haskell. Собственно, Вам предполагается реализовать небольшой фреймворк для таких вычислений.

#### Задача

1. Реализовать свой тип комплексных чисел. Подумайте, как это правильно сделать с учетом Ваших текущих знаний. Тип должен являться составным, т.е. должен принимать другой тип в качестве конструктора типа. Подумайте, зачем делать именно так
2. Вам необходимо реализовать следующие вещи:
  - 1) Квантовое состояние: определяется комплексным числом и строковой меткой. Также оно является функтором.
  - 2) Кубит: список квантовых состояний. Необходимо реализовать следующие функции:

```
toList::Qubit a -> [Complex a]
toLabelList::Qubit a->[String]
fromList:: [Complex a]->[String]->Qubit a
toPairList:: Qubit a->[(Complex a,String)]
fromPairList:: [(Complex a,String)] -> Qubit a
scalarProduct:: Qubit a ->Qubit a ->a
entangle::Qubit a ->Qubit a ->Qubit a
```

(сцепление двух кубитов в новую квантовую систему) (тензорное произведение двух кубитов в векторном представлении)(попарное перемножение всех состояний первых двух кубитов, новой амплитудой является произведение амплитуд, а муткой - конкатенация меток) (вот теперь должно быть понятно)
3. Обязательным является требование самостоятельно в нужных функциях указывать ограничения на тип-параметр.
4. Самостоятельно реализовать набор джентльмена для ваших типов - Ord,Eq, Show.