

Летучка 6 · 7 минут · без конспектов · Ваше ФИ: _____

1. `data Choice = A Int | B Int Int | C`.
Результат `case B 1 2 of { A x -> x + 3; B x y -> x + y; C -> 42 }`: _____
2. Объявление `Either`: _____
3. Существует ли значение типа `Maybe` без типового аргумента? _____
4. Наиболее общий тип `Left 42`: _____
5. Результат `map not [True, False]`: _____

✂ _____

Листок 6. Классы типов

Задание 1 · слайд 5. Допишите инстанс: конструкторы отображаются строками "ERROR", "WARNING", "DEBUG", "INFO".

```
data LogLevel = Error | Warning | Debug | Info

instance Show LogLevel where
  show Error    = "ERROR"
  show Warning  = _____
  show Debug    = _____
  show Info     = _____
```

Задание 2 · слайд 6. Допишите ограничение в сигнатуре. Предскажите результат вызова.

```
showWithPrefix :: _____ => a -> String -> String
showWithPrefix x prefix = prefix ++ show x

showWithPrefix Warning "level: "
```

Задание 3 · слайд 7. Напишите целиком инстанс `Show` для пары `(a, b)`, включая необходимые ограничения. Формат результата: `"(True,3)"` для `(True, 3 :: Int)`.

Задание 4 · слайд 8. Придумайте и напишите целиком класс для парсинга данных `MyRead`. Его метод должен принимать строку и возвращать разобранное значение либо отсутствие результата при неудаче.

Задание 5 · слайд 9. Напишите сигнатуру и реализацию `allValues`, перебирающей все значения конечного перечислимого типа. Какие классы нужны для границ и перехода к следующему значению?

Задание 6 · слайд 10. Реализуйте проверку равенства функций: напишите целиком инстанс `Eq` для `(a -> b)` с необходимыми ограничениями. Область входов конечная и небольшая; используйте `allValues`.

Задание 7 · слайд 16. Допишите инстанс списка по его конструкторам.

```
instance Functor [] where
  fmap f []      = _____
  fmap f (x:xs) = _____
```

Задание 8 · слайд 17. Определите кайнды `(,)`, `(->)` и `(,)` `Int`. Реализуйте функтор для пары.