

Задание №1.

```
module Homework2_1 where
```

```
data Complex = Complex {realPart :: Float, imagPart :: Float} deriving (Eq, Show)
```

```
instance Num Complex where
```

```
Complex a_real a_img + Complex b_real b_img = Complex (a_real + b_real) (a_img + b_img)
Complex a_real a_img - Complex b_real b_img = Complex (a_real - b_real) (a_img - b_img)
Complex a_real a_img * Complex b_real b_img = Complex (a_real * b_real - a_img * b_img)
(a_real * b_img + a_img * b_real)
negate (Complex a_real a_img) = Complex ((-1) * a_real) ((-1) * a_img)
abs (Complex a_real a_img) = Complex ((a_real ** 2 + a_img ** 2)**(1/2)) 0
signum (Complex a_real a_img) = Complex (a_real/((a_real ** 2 + a_img ** 2)**(1/2))) (a_img/
((a_real ** 2 + a_img ** 2)**(1/2)))
```

Задание №2.

```
module Fun where
```

```
newtype Fun a b = Fun {getFun :: a -> b}
```

```
instance Functor (Fun a) where
```

```
  fmap f (Fun b) = Fun(\x -> f (b x))
```

```
  -- instance Functor ((->) a) where
```

```
  -- fmap f g = f . g
```

```
instance Applicative (Fun a) where
```

```
  pure a = Fun(\_ -> a)
```

```
  Fun a <*> Fun b = Fun(\x -> (a x) (b x))
```

```
  -- instance Applicative ((->) a) where
```

```
  -- pure x = (\_ -> x)
```

```
  -- f <*> g = \x -> f x (g x)
```