TYPES WITH CLASS

WIM VANDERBAUWHEDE

Type classes are a way to overload functions or operators by putting constraints on polymorphism.

For example, we have seen that:

```
(+) :: a -> a -> a
```

is not OK because we want to restrict addition to numbers.

Likewise,

```
(<) :: a -> a -> Bool
```

is not OK because it is not clear a priory how to compare to arbitrary types.

To address this issue Haskell provides type classes. These restrict the polymorphism. For example:

```
(+) :: Num a => a -> a -> a
```

says that the type a must be a numeric type, and

```
(<) :: Ord a => a -> a -> Bool
```

says that a must be orderable.