

5.6 Dimensions and units

Properties of physical entities are as mentioned often represented by value objects. Such properties are often called *quantities*. A quantity may represent properties like length, mass, speed and velocity of a physical entity like a vehicle. A quantity can be denoted by measurement that consist of a number (called the *magnitude* of the measurement) and a *unit*. As an example 12 centimeters is a measurement of a quantity length where the magnitude is 12 and the unit is centimeters.

So far we have used primitive values like float to represent the *magnitude* (value) of a quantity. A quantity, however, also has an associated unit. It is thus important to be able to represent the unit being associated with a given quantity.

Consider a physical entity like a car. It has properties (quantities) length and weight. The length may be measured in centimeter, and its weight may be measured in kilograms. If we just use a float we are not able to distinguish between values of these two properties:

```
class Car:
  l: var float
  w: var float
```

There is nothing that prevents us from assigning l to w (`w := l`) or compare l and w (`l = w`) since the compiler only knows that l and w are two floats.

To handle this, we may specify the unit of l and w as follows:

```
class Car:
  l: var Length.Centimeter
  w: var Mass.Kilogram
```

Here Length is a singular object representing the dimension length as defined by the [International System of Units \(SI\)](#). Length defines local types (classes) representing units like Millimeter, Centimeter, Meter, etc.

Mass is a singular object that in a similar way represents the dimension mass as defined by SI, and it has types like Kilogram, Gram, etc.

When l and w are declared as above, the compiler prevents assignments like `w := l` since w and l have different types. And a comparison like `l = w` is also prevented.

The types Centimeter and Kilogram include a float representing the numerical value of the magnitude of the quantity as well as the unit of the quantity.

Data-items representing quantities may be assigned a numerical value, but a unit has to be supplied:

```
l := 3.1 m
w := 12 kg
```

The variable l as assigned 3.1 meter and w is assigned 12 kilogram.

Addition and subtraction of quantities gives a new quantity of the same dimension. Multiplication, division and exponentiation give a derived dimension. Consider the following example:

```

l1,l2,l3: var Length.Meter
a: var Area.SquareMeter
l1 := 12 m
l2 := 3 m
l3 := l1 + l2 -- gives l3 = 15 m
a := l1 * l2 -- gives a = 36 m2 (squaremeter)

```

It is beyond the scope of this book to explain SI in details, but if you have to work with units we recommend that you study the SI. Here we just summarise some of the overall characteristics of SI:

According to the SI, there are seven base dimensions: *time*, *length*, *mass*, *electric current*, *thermodynamic temperature*, *amount of substance*, and *luminous intensity*. There are seven base units, one per dimension, and each base unit has an associated symbol. The units are a.o. the *second* (the unit of *time* with the symbol *s*), *meter* (*length*, *m*), and *kilogram* (*mass*, *kg*). In addition to the base units each dimension may have a number of other units. Meter is the base unit of length, but other units are centimeter and millimeter.

In addition, there is an unlimited number of additional units, called *derived units*, like *area*, *velocity*, and *force*. A derived unit can always be represented as products and/or powers of the base units. The unit of area is *square meter* defined as $length \cdot length$; the unit of velocity is $length \cdot time^{-1}$; etc.

The qBeta library includes objects representing several dimensions and associated units as defined in SI. These objects are modules as described in chapter . The table below shows some of the dimensions and units supported by qBeta.

<i>Dimension</i>	<i>Unit</i>	<i>Module name</i>
time	second	Time
length	meter	Length
mass	kilogram	Mass
electric current	ampere	ElectricCurrent
thermodynamic temperature	kelvin	ThermodynamicTemperature
amount of substance	mole	AmountOfSubstance
luminous intensity	candela	LuminousIntensity
area	square meter	Area
volume	cubic meter	Volume
velocity	meter per second	Velocity
acceleration	meter per second squared	Acceleration
force	newton	Force