Table of Contents

MoonBit QuickCheck Package

MoonBit QuickCheck package provides property-based testing capabilities by gener
ating random test inputs.

Basic Usage

Generate random values of any type that implements the Arbitrary trait:

```
1
2   test "basic generation" {
3     let b : Bool = @quickcheck.gen()
4     inspect(b, content="true")
5     let x : Int = @quickcheck.gen()
6     inspect(x, content="0")
7
8
9     let sized : Array[Int] = @quickcheck.gen(size=5)
10    inspect(sized.length() <= 5, content="true")
11  }
```

Multiple Samples

Generate multiple test cases using the samples function:

```
1
2   test "multiple samples" {
3     let ints : Array[Int] = @quickcheck.samples(5)
4     inspect(ints, content="[0, 0, 0, -1, -1]")
5     let strings : Array[String] = @quickcheck.samples(12)
6     inspect(
7       strings[5:10],
8       content=(
9         #|["E\b\u{0f} ", "", "K\u{1f}[", "!@", "xvLxb"]
10      ),
11    )
12  }
```

Built-in Types

QuickCheck provides Arbitrary implementations for all basic MoonBit types:

```
1
2   test "builtin types" {
3
4     let v : (Bool, Char, Byte) = @quickcheck.gen()
5     inspect(v, content="(true, '#', b'\\x12')")
6
7     let v : (Int, Int64, UInt, UInt64, Float, Double, BigInt) = @quickchec
8     inspect(v, content="(0, 0, 0, 0, 0.1430625319480896, 0.330984466952546
9
10
11    let v : (String, Bytes, Iter[Int]) = @quickcheck.gen()
12    inspect(
13      v,
14      content=(
15        #|("", b"", [])
16      ),
17    )
18  }
```

Custom Types

Implement Arbitrary trait for custom types:

```
1
2   struct Point {
3     x : Int
4     y : Int
5   } derive(Show)
6
7
8   impl Arbitrary for Point with arbitrary(size, r0) {
9     let r1 = r0.split()
10    let y = @quickcheck.Arbitrary::arbitrary(size, r1)
11    { x: @quickcheck.Arbitrary::arbitrary(size, r0), y }
12  }
13
14
15  test "custom type generation" {
16    let point : Point = @quickcheck.gen()
17    inspect(point, content="{x: 0, y: 0}")
18    let points : Array[Point] = @quickcheck.samples(10)
19    inspect(
20      points[6:],
21      content="[{x: 0, y: 1}, {x: -1, y: -5}, {x: -6, y: -6}, {x: -1, y: 7
22    )
23  }
```

The package is useful for writing property tests that verify code behavior acros
s a wide range of randomly generated inputs.