KEYWORDS

***actor***

***as*** // conversion of a value to another Type (can raise error)

***be*** // behavior, works asynchronous

***box*** // default *reference capability* – object must be readable, but not writable

***break*** // to step out of a loop statement

***class***

***compile\_error***// will provoke a compile error

***continue*** // loop statement

***consume*** // move a value to a new variable, leaving the original variable empty

***do*** // loop statement

// after a with statement

***else*** // conditional statement (in if, for, while, repeat, try (as a catch block), match)

***elseif*** // conditional statement, also used with ifdef

***end*** // ending of (if then, ifdef, while do, for in, repeat until, try, object, lambda, recover, match)

***error*** // raises an error

***for*** // loop statement

***fun*** // function, works synchronous

***if*** // (1) conditional statement

// 2) when specifying a guard in a pattern match

***ifdef*** // when defining a build flag at compiling: ponyc –D “foo” : ifdef “foo” == true

***in*** // loop statement

***interface*** // used in structural subtyping

***is*** // used in nominal subtyping, class CCC is TTT (TTT is a trait) // also used in structural subtyping

// in type aliasing

***lambda*** // to make a closure

***let*** // declaration of immutable variable

***match*** // pattern matching

***new*** // constructor

***var*** // declaration of mutable variable

***object*** // to make an object literal

***primitive*** // declares a primitive type

***recover*** // take away the reference capability of a variable (to get an iso to pass to // another actor)

***ref*** // *reference capability* –

// reference type = object (on which function is called) is mutable

// fun ref: a function that changes internal state (fields)

***repeat*** // loop statement

***return*** // to return early from a function

***then*** // in if conditional statement

// as a (finally) block in try

***this*** // the current object

***trait*** // used in nominal subtyping

***try*** // error handling

***type*** // to declare a type alias:

***until*** // loop statement

***use*** // 1) using a package

// 2) using an external library foo: use "**lib**:foo**"**

// 3) declaration of an FFI signature

// 4) add a search path for external libraries: use "**path**:/usr/local/lib**"**

***var*** // mutable variable

***val***

***where*** // 1) when specifying named arguments

***while*** // loop statement

***with***// with var = … as Type do

things in a `with` are automatically disposed