The Iris Language Reference Manual

Valerie Zhang, Josh Kim, Ayda Aricanli, Tim Valk, and Trevor Sullivan

October 2023

Contents

1 Introduction

Iris is a general-purpose, object-oriented programming language that combines features from Java, C, and C++. The language is class-oriented in that *almost* everything is a class. As such, it supports inheritance as well as other features like polymorphic, mutable lists, and encapsulation.

Iris introduces a spin on the normal conventions of encapsulation by introducing `permit` class members. These members are accessible by a class's sub-classes or other classes in its private collection of class names `permitted`. However, unlike *friend* classes in C++, a class's permitted classes may only access the `permit` methods of the class rather than all class methods.

1.1 Notation

Code is distinguished from prose by using this `font`.

2 Lexical Conventions

2.1 Comments

The characters `..` introduce a single-line comment.
The characters `.~*` introduce a multi-line comment, which terminates with `*~.`

2.2 Identifier Conventions

Any valid identifier in Iris is an uppercase or lowercase alphabetic character followed by any number of uppercase or lowercase alphabetic characters and/or digits. Uppercase and lowercase letters are distinct. The underscore _ is also an alphabetic character.

2.3 Keywords

`if`	`public`	`bool`
`else`	`permit`	`char`
`for`	`permitted`	`string`
`while`	`private`	`float`
`return`	`void`	`List`
`class`	`univ`	`Olympus`
`new`	`const`	
`of`	`int`	

2.4 Literals

Iris allows for literals for each primitive data type:

2.4.1 Integer literals

An integer literal is any sequence of one or more digits. Integer literals can begin with exactly one "-", and if so, are assumed to be negative. All integer literals are assumed to be in base 10.

2.4.2 Float literals

A floating literal consists of 3 parts: a sequence of one or more digits, the decimal point, and another sequence of one or more digits. All 3 of these parts MUST be present for a float literal. Floating literals can optionally begin with exactly one "`-`", and if so, are assumed to be negative. All float literals are assumed to be in base 10.

2.4.3 Boolean literals

`true` and `false` are the only two boolean literals. These literals can be constructed from ASCII characters.

2.4.4 Character literals

A character literal consists of one or two characters encapsulated in `'`. Inside the single quotes, there can be a single ASCII character, or one of the following special cases occurs: one of `'`, `n`, or \ is preceded by a \ to encode the single quote, newline, or backslash character, respectively.

2.4.5 String literals

A string literal consists of a series of character literals surrounded by double quotes. Within the string literal, a `"` character must be preceded by a single \.

3 What's In a Name?

optional univ, datatype, identifier,

4 Types

4.1 Operators

5 Expressions

6 Classes

6.1 Encapsulation

6.2 Inheritance

7 Built-In Functions

7.1 Lists

7.2 The Olympus Class

8 Scope