alxLang whitepaper

Donatas Mockus

September 16, 2023

1 Tokens

1.1 Operators

1.1.1 Binary

```
  // Binary
  // +, -, *, /, ^, %, :, <<, >>, ==, !=
  // Assignment
  // =, +=, -=, *=, /=, ^=, %=, :=, <<=, >>=, ===
```

1.1.2 Unary

```
  // !, ++, --
```

1.2 Keywords

```
  // Number types
  bool, char, int, uint, long, ulong, float, double
  // Branching
  if, else, while, for, do, break, continue
  // Objects
  class, struct, interface, public, private, protected, final,
  virtual, override, partial
  // Other
  return
```

2 Safety

2.1 What is safety?

The safety of a language can be defined as follows:

- Memory safety
- Type safety
- Resource safety
- Bounds checking

[1]Not an exhaustive list
The following subsections will explore how the language attempts to ensure these principles.

2.2 Memory safety

No raw pointers
All pointers are shared pointers by default

2.3 Type safety

2.4 Resource safety

RAII

2.5 Bounds checking

3 OOP

Both classes and structs may have private, protected, and public member variables.

3.1 Structs

Structs must:

- Be trivially copyable
- Have contiguous memory
- Not contain virtual methods

3.2 Classes

4 Example Code

Hello, world

```
using stdio;
/* I'm a block comment */
int main(string argv[]) // I'm a line comment
{
  const world = "world";
  println($"Hello {}", world);
}
```

Classes, interfaces, and inheritance

```
import stdio;

namespace Animals
{
interface IAnimal {
  string Name { get; private set; }
  int Age { get; private set; }
}

class Cat : IAnimal {
public:
  string Name { get; private set; }
  int Age { get; private set; }

  Cat(string name, int age) : Name(name), Age(age) // Constructor
  {
    println("Created {} which is {} old!", Name, Age);
  }


private:
  ~Cat() // Private destructors allow GC to manage lifetime of the object.
  {
    println("{Name} has been destructed");
  }
}

class NorweigianForset final : Cat {
public:
  void Meow() { println("Meow); }
}

}
using namespace Animals;
int main()
{
  Animals.IAnimal tuxie = new Animal.Cat("Tuxie", 6);
  NorweigianForset pepper = new("Pepper", 4);
  pepper.Meow();
}
```

Cool keywords

```
  deprecated class StringView { ... }

  int main()
  {
    StringView str = new(); // Will throw a deprecation warning.
  }


  // A.alx
  partial class PartialClass { ... }
  // B.alx
  partial class PartialClass { ... }
```