Vince Alihan
4-19-2024
CS 420

Assignment 6

Name: Kat

Overview:

The name of my language is Kat, short for Kathleen, my girlfriend whom I am designing this language for. The purpose of this language is to make programming easier for cat-lovers and for her.

Purpose:

To make programming easier and “cuter” in Kat’s words.

Target Domain:

My girlfriend is the target domain who I want to design this language for. She absolutely hates programming, but loves cats. So, why not combine those 2 aspects, to turn something she hates, into something more palatable to her.

Design:

Syntax:

- This programming language is based on Java and some aspects of Python, and is a high level, class-based, object oriented language designed to be “cuter”.
- Programs are created similar to Java, with the need to create a class then a main method.
- Rather than using a semicolon though, since she sometimes forgets them, you won’t need one, instead, the program will rely on whitespace to determine if code belongs to a certain block, similar to Python.
- If a syntax error occurs, rather than print out big scary red text, words of affirmation will print out and encourage the programmer to keep trying.
- All classes will be dynamic, and not static, similar to Python.
- Must have a main method named “scratcher”
- Syntax is nearly similar to Java, but the semantics are where things differ.

Semantics:

- *Classes/methods* will be declared with any identifier, but may have a modifier of either:

  floofyblackones, purrivate, britishshorthair, depawult
  (public) (private) (protected) (no access modifier)
  Or none at all.
  These modifiers also apply to any variable using them.
- *Variables* will be declared with any identifier, but must have a datatype of either:

  biscuits, yarn, boolean
  (number) (string) (she liked the name boolean)
  To declare an int variable, the structure is:

```
I want 5 biscuits in a bag! //value of 5 stored into variable bag
```

String:

```
I want "String" yarn in a bag! //"String" literal
```

Boolean:

```
it gotta be True //true is stored in variable it
```

Note that all variable declarations end with an exclamation mark, and all declarations depending on type must contain the underlined phrases.

- *Methods* will have return types with data types of biscuits, yarn, boolean, or void.
- *Conditional Statements* will remain the same structure as Java, but will use "eats" to replace "purr", "purrsif" to replace "else if", and "hiss" for "else". Conditional Statements in Kat also use Python logic operators such as and, or, not, which Kat ended up liking more.

```
purr (i % 3 == 0) or (i % 5 == 0):
purrs if (i % 3 == 0):
hiss:
```

- *Loops* will not be your typical Java loops, but instead will be jump/goto statements that jump to labels. To declare one, the structure is:

```
mice:
pounce on mice (condition)
```

- *Print* statements will always start with the word "Meow". The structure is

```
Meow "Hello World"
```

Sample Program:

```
floofyblackones class prog1
    floofyblackones void scratcher()
        Meow "Hello World"

class prog2
    void scratcher()
        I want 10 biscuits in a var1!  //value of 10 stored in var1
        purr (var1 == 10):
            Meow var1
        hiss:
            Meow "Not enough"

orangecat class prog3
    flooyblackones void scratcher()
        I want 0 biscuits in a i!
        mice
            Meow i
            i++
        pounces on mice (i < 5)
        purr (i < 5):
            pounce label
        hiss
            pounce end
```

Ecosystem:

In the future, since this language adopts structures from other languages, it cannot be interpreted in any current IDE, so we will create our own, just for Kat.

Challenges:

Those who dislike cats and are okay with typical programming structures and programming in general, probably wouldn’t want to use this language.