Comparison of Object-Oriented Programming Languages

Timothy Clark (488232)

April 28, 2008

Introduction

What is object orientation?

Comparison Variables

Language Comparison

- Visual Basic
- Java
- Python

Conclusion

Introduction

- Looking at programming languages

Introduction

- Looking at programming languages
- Everyone has their favourite language

Introduction

- Looking at programming languages
- Everyone has their favourite language
- Object oriented languages are quite popular

Introduction

- Looking at programming languages
- Everyone has their favourite language
- Object oriented languages are quite popular
- Choose the right tool for the job

What is object orientation?

- Type of programming language

What is object orientation?

- Type of programming language
- Characterised by:

What is object orientation?

- Type of programming language
- Characterised by:
  - Inheritance

What is object orientation?

- Type of programming language
- Characterised by:
  - Inheritance
  - Objects

What is object orientation?

- Type of programming language
- Characterised by:
  - Inheritance
  - Objects
  - Classes

What is object orientation?

- Type of programming language
- Characterised by:
  - Inheritance
  - Objects
  - Classes
  - Encapsulation

What is object orientation?

- Type of programming language
- Characterised by:
  - Inheritance
  - Objects
  - Classes
  - Encapsulation
  - Methods

What is object orientation?

- Type of programming language
- Characterised by:
  - Inheritance
  - Objects
  - Classes
  - Encapsulation
  - Methods
  - Message Passing

What is object orientation?

- Type of programming language
- Characterised by:
  - Inheritance
  - Objects
  - Classes
  - Encapsulation
  - Methods
  - Message Passing
  - Polymorphism

What is object orientation?

- Type of programming language
- Characterised by:
  - Inheritance
  - Objects
  - Classes
  - Encapsulation
  - Methods
  - Message Passing
  - Polymorphism
  - Abstraction

What is object orientation?

- Type of programming language
- Characterised by:
  - Inheritance
  - Objects
  - Classes
  - Encapsulation
  - Methods
  - Message Passing
  - Polymorphism
  - Abstraction
- Designed to make code reuse and designing large systems easier

Comparison Variables

- Hard to decide how to compare languages

Comparison Variables

- Hard to decide how to compare languages
- I've looked at:

Comparison Variables

- Hard to decide how to compare languages
- I've looked at:
  - Running speed:

Comparison Variables

- Hard to decide how to compare languages
- I've looked at:
  - Running speed:
    - Fully Compiled

Comparison Variables

- Hard to decide how to compare languages
- I've looked at:
  - Running speed:
    - Fully Compiled is faster than
    - Half compiled(Compiled to byte code)

Comparison Variables

- Hard to decide how to compare languages
- I've looked at:
  - Running speed:
    - Fully Compiled is faster than
    - Half compiled(Compiled to byte code) is faster than
    - Fully interpreted

Comparison Variables

- Hard to decide how to compare languages
- I've looked at:
  - Running speed:
    - Fully Compiled is faster than
    - Half compiled(Compiled to byte code) is faster than
    - Fully interpreted
  - Portability:

Comparison Variables

- Hard to decide how to compare languages
- I've looked at:
  - Running speed:
    - Fully Compiled is faster than
    - Half compiled(Compiled to byte code) is faster than
    - Fully interpreted
  - Portability:
    - Interpreted

Comparison Variables

- Hard to decide how to compare languages
- I've looked at:
  - Running speed:
    - Fully Compiled is faster than
    - Half compiled(Compiled to byte code) is faster than
    - Fully interpreted
  - Portability:
    - Interpreted or Half Complied

Comparison Variables

- Hard to decide how to compare languages
- I've looked at:
  - Running speed:
    - Fully Compiled is faster than
    - Half compiled(Compiled to byte code) is faster than
    - Fully interpreted
  - Portability:
    - Interpreted or Half Complied is more portable than
    - Compiled

Comparison Variables

- Hard to decide how to compare languages
- I've looked at:
  - Running speed:
    - Fully Compiled is faster than
    - Half compiled(Compiled to byte code) is faster than
    - Fully interpreted
  - Portability:
    - Interpreted or Half Complied is more portable than
    - Compiled
    - Changes how easy it is to move platforms

Comparison Variables

- Hard to decide how to compare languages
- I've looked at:
  - Running speed:
    - Fully Compiled is faster than
    - Half compiled(Compiled to byte code) is faster than
    - Fully interpreted
  - Portability:
    - Interpreted or Half Complied is more portable than
    - Compiled
    - Changes how easy it is to move platforms
  - Type Strength

Comparison Variables

- Hard to decide how to compare languages
- I've looked at:
  - Running speed:
    - Fully Compiled is faster than
    - Half compiled(Compiled to byte code) is faster than
    - Fully interpreted
  - Portability:
    - Interpreted or Half Complied is more portable than
    - Compiled
    - Changes how easy it is to move platforms
  - Type Strength
  - What the language solves

Comparison Variables

- Hard to decide how to compare languages
- I've looked at:
  - Running speed:
    - Fully Compiled is faster than
    - Half compiled(Compiled to byte code) is faster than
    - Fully interpreted
  - Portability:
    - Interpreted or Half Complied is more portable than
    - Compiled
    - Changes how easy it is to move platforms
  - Type Strength
  - What the language solves
  - How popular it is

Visual Basic

- Microsoft

Visual Basic

- Microsoft
- Half Compiled (average speed)

Visual Basic

- Microsoft
- Half Compiled (average speed)
- Only compiles for Windows

Visual Basic

- Microsoft
- Half Compiled (average speed)
- Only compiles for Windows
- Supports both strong and loose typing depending on compile options.

Visual Basic

- Microsoft
- Half Compiled (average speed)
- Only compiles for Windows
- Supports both strong and loose typing depending on compile options.
- Makes making GUI programs and prototypes for Windows very easy

Java

- Sun

Java

- Sun
- Half Compiled (average speed)

Java

- Sun
- Half Compiled (average speed)
- Runs in a virtual machine that Sun provide for lots of platforms

Java

- Sun
- Half Compiled (average speed)
- Runs in a virtual machine that Sun provide for lots of platforms
- Strongly Typed:

Java

- Sun
- Half Compiled (average speed)
- Runs in a virtual machine that Sun provide for lots of platforms
- Strongly Typed:
  - Explicit conversions required

Java

- Sun
- Half Compiled (average speed)
- Runs in a virtual machine that Sun provide for lots of platforms
- Strongly Typed:
  - Explicit conversions required
  - More compile errors

Java

- Sun
- Half Compiled (average speed)
- Runs in a virtual machine that Sun provide for lots of platforms
- Strongly Typed:
    - Explicit conversions required
    - More compile errors
    - Less run-time errors

Java

- Sun
- Half Compiled (average speed)
- Runs in a virtual machine that Sun provide for lots of platforms
- Strongly Typed:
  - Explicit conversions required
  - More compile errors
  - Less run-time errors
- Like C

Java

- Sun
- Half Compiled (average speed)
- Runs in a virtual machine that Sun provide for lots of platforms
- Strongly Typed:
  - Explicit conversions required
  - More compile errors
  - Less run-time errors
- Like C
- Large class library

Java

- Sun
- Half Compiled (average speed)
- Runs in a virtual machine that Sun provide for lots of platforms
- Strongly Typed:
    - Explicit conversions required
    - More compile errors
    - Less run-time errors
- Like C
- Large class library
- Good for teaching

Python

- Python Software Foundation

Python

- Python Software Foundation
- Fully interpreted (slow)

Python

- Python Software Foundation
- Fully interpreted (slow)
- Syntax Checker

Python

- Python Software Foundation
- Fully interpreted (slow)
- Syntax Checker
- Standard interpreter available for most common platforms

Python

- Python Software Foundation
- Fully interpreted (slow)
- Syntax Checker
- Standard interpreter available for most common platforms
- Easy to port interpreter:

Python

- Python Software Foundation
- Fully interpreted (slow)
- Syntax Checker
- Standard interpreter available for most common platforms
- Easy to port interpreter:
  - Source code available for interpreter

Python

- Python Software Foundation
- Fully interpreted (slow)
- Syntax Checker
- Standard interpreter available for most common platforms
- Easy to port interpreter:
  - Source code available for interpreter
  - Fully comprehensive language specification

Python

- Python Software Foundation
- Fully interpreted (slow)
- Syntax Checker
- Standard interpreter available for most common platforms
- Easy to port interpreter:
  - Source code available for interpreter
  - Fully comprehensive language specification
- Technically strongly typed:

Python

- Python Software Foundation
- Fully interpreted (slow)
- Syntax Checker
- Standard interpreter available for most common platforms
- Easy to port interpreter:
  - Source code available for interpreter
  - Fully comprehensive language specification
- Technically strongly typed:
  - Types automatically

Python

- Python Software Foundation
- Fully interpreted (slow)
- Syntax Checker
- Standard interpreter available for most common platforms
- Easy to port interpreter:
  - Source code available for interpreter
  - Fully comprehensive language specification
- Technically strongly typed:
  - Types automatically
  - Type clashes cause run time errors

Python

- Python Software Foundation
- Fully interpreted (slow)
- Syntax Checker
- Standard interpreter available for most common platforms
- Easy to port interpreter:
  - Source code available for interpreter
  - Fully comprehensive language specification
- Technically strongly typed:
  - Types automatically
  - Type clashes cause run time errors
- Designed to:

Python

- Python Software Foundation
- Fully interpreted (slow)
- Syntax Checker
- Standard interpreter available for most common platforms
- Easy to port interpreter:
    - Source code available for interpreter
    - Fully comprehensive language specification
- Technically strongly typed:
    - Types automatically
    - Type clashes cause run time errors
- Designed to:
    - Make it easy to read and program

Python

- Python Software Foundation
- Fully interpreted (slow)
- Syntax Checker
- Standard interpreter available for most common platforms
- Easy to port interpreter:
    - Source code available for interpreter
    - Fully comprehensive language specification
- Technically strongly typed:
    - Types automatically
    - Type clashes cause run time errors
- Designed to:
    - Make it easy to read and program
    - Have a big default library

Python

- Python Software Foundation
- Fully interpreted (slow)
- Syntax Checker
- Standard interpreter available for most common platforms
- Easy to port interpreter:
  - Source code available for interpreter
  - Fully comprehensive language specification
- Technically strongly typed:
  - Types automatically
  - Type clashes cause run time errors
- Designed to:
  - Make it easy to read and program
  - Have a big default library
- Commonly used as scripting language

Conclusion

- Varying processing speed

Conclusion

- Varying processing speed
- Platform independence varies:

Conclusion

- Varying processing speed
- Platform independence varies:
    - Visual Basic locked to just windows

Conclusion

- Varying processing speed
- Platform independence varies:
  - Visual Basic locked to just windows
  - Python works (or can be made to work) on any platform

Conclusion

- Varying processing speed
- Platform independence varies:
    - Visual Basic locked to just windows
    - Python works (or can be made to work) on any platform
- Java is the only one that is strongly typed

Conclusion

- Varying processing speed
- Platform independence varies:
  - Visual Basic locked to just windows
  - Python works (or can be made to work) on any platform
- Java is the only one that is strongly typed
- Good at different things:

Conclusion

- Varying processing speed
- Platform independence varies:
  - Visual Basic locked to just windows
  - Python works (or can be made to work) on any platform
- Java is the only one that is strongly typed
- Good at different things:
  - Visual Basic is good for windows prototyping

Conclusion

- Varying processing speed
- Platform independence varies:
  - Visual Basic locked to just windows
  - Python works (or can be made to work) on any platform
- Java is the only one that is strongly typed
- Good at different things:
  - Visual Basic is good for windows prototyping
  - Java is good for larger structured programs

Conclusion

- Varying processing speed
- Platform independence varies:
  - Visual Basic locked to just windows
  - Python works (or can be made to work) on any platform
- Java is the only one that is strongly typed
- Good at different things:
  - Visual Basic is good for windows prototyping
  - Java is good for larger structured programs
  - Python is a good platform independent prototyping, scripting and as a "glue" language

Conclusion

- Varying processing speed
- Platform independence varies:
  - Visual Basic locked to just windows
  - Python works (or can be made to work) on any platform
- Java is the only one that is strongly typed
- Good at different things:
  - Visual Basic is good for windows prototyping
  - Java is good for larger structured programs
  - Python is a good platform independent prototyping, scripting and as a "glue" language
- Downfalls:

Conclusion

- Varying processing speed
- Platform independence varies:
  - Visual Basic locked to just windows
  - Python works (or can be made to work) on any platform
- Java is the only one that is strongly typed
- Good at different things:
  - Visual Basic is good for windows prototyping
  - Java is good for larger structured programs
  - Python is a good platform independent prototyping, scripting and as a "glue" language
- Downfalls:
  - Visual Basic encourages sloppy programming practice

Conclusion

- Varying processing speed
- Platform independence varies:
  - Visual Basic locked to just windows
  - Python works (or can be made to work) on any platform
- Java is the only one that is strongly typed
- Good at different things:
  - Visual Basic is good for windows prototyping
  - Java is good for larger structured programs
  - Python is a good platform independent prototyping, scripting and as a "glue" language
- Downfalls:
  - Visual Basic encourages sloppy programming practice
  - Java is harder to write programs in that the other two

Conclusion

- Varying processing speed
- Platform independence varies:
  - Visual Basic locked to just windows
  - Python works (or can be made to work) on any platform
- Java is the only one that is strongly typed
- Good at different things:
  - Visual Basic is good for windows prototyping
  - Java is good for larger structured programs
  - Python is a good platform independent prototyping, scripting and as a "glue" language
- Downfalls:
  - Visual Basic encourages sloppy programming practice
  - Java is harder to write programs in that the other two
  - Python is slow compared to the other two and can have more run-time typing errors

When would I use these?

- Visual Basic for fast Windows GUI prototyping

When would I use these?

- Visual Basic for fast Windows GUI prototyping
- Python for scripting, and prototyping on non windows platforms

When would I use these?

- Visual Basic for fast Windows GUI prototyping
- Python for scripting, and prototyping on non windows platforms
- Java for programming in larger more structured programs

Finally

- These slides are available at http://sucs.org/˜eclipse

Finally

- These slides are available at http://sucs.org/˜eclipse
- Any Questions?