

基于 Java 与 Python 的面向对象编程的基本特征研究

Research on the Basic Features of Object Oriented Programming Based on Java and Python

丁烈骁

Liexiao Ding

美国大数据分析交易公司 美国 · 30067

Big Data Analytics Trading Company in the United States, 30067, USA

摘要: 论文以“Java 与 Python”两种主流编程语言为研究对象,分析和对比不同种类对象编程语言在继承与多态之间的差异性,发现 Java 仅仅支持单继承;Python 可以支持多继承。Java 编程语言通过接口设计的方式,可以间接性地进行多继承。此外,分析了 Java 垃圾回收机制的意义、主 GC 触发条件,并将其与 Python 编程语言进行对比和分析。最后,分析 Java 内存概况和结构。希望通过这次研究,为相关人员提供有效的借鉴和参考。

Abstract: This paper takes “Java and Python” as the research object, analyzes and compares the differences between inheritance and polymorphism in different types of object programming languages, and finds that Java only supports single inheritance; Python can support multiple inheritance. The Java programming language can indirectly perform multiple inheritance through interface design. In addition, the significance of the Java garbage collection mechanism and the triggering conditions of the main GC were analyzed, and compared and analyzed with the Python programming language. Finally, analyze the overview and structure of Java memory. I hope to provide effective reference and guidance for relevant personnel through this study.

关键词: 面向对象编程; Java; Python; 继承与多态

Keywords: object-oriented programming; Java; Python; inheritance and polymorphism

DOI: 10.12346/csai.v2i1.9106

1 引言

进行程序设计中,主要用到面向过程编程思想和面向对象编程思想。其中,面向过程编程思想主要是指借助函数,以线性步骤为载体,对程序进行设计,整个设计过程比较复杂、繁琐,需要开发人员具备扎实的编程基础^[1]。运用这种编程思想的语言以 PLC 语言为代表;面向对象编程思想主要借助对象指类,将对象视为程序基本单元,使得软件表现出较高的灵活性。面向对象具有继承、封装和多态三大特征^[2]。Java 与 Python 作为两种不同面向对象编程语言,目前被发展为主流语言,该语言具有较高的研究价值。因此,强化对 Java 与 Python 两种不同面向对象编程语言的特征研究显得尤为重要。

2 Java 与 Python 的多态与继承对比

2.1 多态

多态作为一种常见概念,主要存在于面向对象语言里。

多态主要是指同时存在于同一个程序中的名称相同而内容不同的方法。在 Java 面向对象编程语言中,多态主要划分为以下两大组成部分:①静态。静态主要是指在进行编译时,系统采用重载的方式,确定出需要调用的函数名。②动态。动态主要是指在执行程序时,通过借助虚函数,采用重写的方式,对指针所指向的对象进行动态化确定^[3]。面向对象编程语言在具体编译时,如果要求变量数据类型是唯一的、确定不变的,那么,该编程语言被称为静态语言,如 Java 语言。反之,被称为动态语言,如 Python 语言。与 Java 静态语言相比,Python 动态语言无需对变量数据类型进行检查和确定,只要方法存在,参数设置正确,就可以对其进行直接调用。例如:在 Python 动态语言中,参数 self 可以实例任何数据类型,只要存在 getName() 方法即可。

2.2 继承

Java 作为一种常用的面向对象编程语言,仅仅支持单继

【作者简介】丁烈骁(1987-),男,中国北京人,硕士,工程师,从事计算机网络架构研究。

承,无法支持多继承,也就是说单个子类无法对应多个父类。单继承好处是可以简化 Java 继承关系,确保单继承关系越来越简单化,便于开发人员高效管理程序。Java 语言要想实现多继承目的,需要利用接口进行实现。Java 语言通过采用多重继承的方式,确保单个子类对多个父类各个功能进行同时继承和获得。Python 作为一种主流的面向对象编程语言,可以直接支持多继承。子类在没有写初始化方法时,可以直接继承首个父类的对象方法^[4]。为保证设计效率,需要采用 Mixin 设计方法,不断地简化 Python 多层次的继承关系,确保 Python 继承关系变得更加简单。

3 Java 的接口

3.1 接口的引入与形式

由于 Java 语言无法支持多继承,因此 Java 程序结构变得更加简单化。但是,支持多继承的 Python 语言相比,Java 单继承的运用,无法清晰有效地表述繁琐复杂的问题。为解决这一问题,Java 语言可以借助接口,对类似于 Python 语言的多继承功能进行实现。接口作为一种数据类型具有一定抽象性,这表明接口无法进行实例化^[5]。因此,在接口中,不存在确定方法和变量,只存在常量和定义好的方法。在进行接口声明时,要将接口视为一个抽象类,只有声明虚函数和静态数据,才能达到声明接口的目的。

3.2 接口的实现

接口这一数据类型具有较高的抽象性,可以避免使用多继承所造成的各种问题。通过借助多个接口,可以对单个类进行实现。例如:Python 语言的多继承功能。接口在具体应用中,其用法与类用法极为类似,尽管无法将 new 操作符直接应用于接口中,但是,可以使用接口,完成对类型名设置。

3.3 接口的多继承

Java 接口多继承如图 1 所示,单个接口可以对多个接口进行同时继承。多继承功能的运用,可以达到合并多个接口规范的目的,同时,还形成程序抽象结构层次。在接口中,所使用的方法仅仅是抽象的原型,因此,有效地解决因重复定义父类方法而导致的重复继承问题,促使程序结构变得越来越简单化。

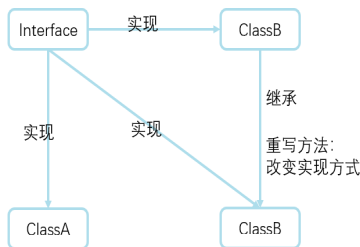


图 1 Java 接口多继承

3.4 接口的应用

通过应用接口,可以精确化、详细化描述类的各项特征。

在 Java 面向对象编程语言中,接口通常被视为“单个类”进行使用,此时,new 操作符无法直接声明接口。在对 Java 程序进行运行时,可以直接确定所调用方法类别,从而达到动态化调用方法的目的。传统程序系统在具体应用时,无法充分发挥和利用 Java 负荷在可再生方面优势,同时,由于网络开放性高、可控性强,导致系统与负荷之间无法建立良好的协调关系。而接口在具体应用时,可以具体化、形象化描述多累特征,可以充分发挥和利用面向对象体系架构在发电方面的优势,并借助多继承功能,确保多个接口同时参与到程序系统负荷调频中,在这样的背景下,主机设备可以执行 95% 以上的数据处理工作。另外,运用 Java 面向对象编程语言,除了可以收集基础性信息外,还能指定相关动作,完全符合网络远程实时控制需求,从而极大地提高用户的使用体验。

4 Java 与 Python 的垃圾回收机制

4.1 Java 垃圾回收机制

4.1.1 垃圾回收的意义

Java 语言主要用到垃圾回收机制,这种机制的运用,可以有效地解决编程内存管理问题,有效地降低开发人员的编程工作量和编程压力。单个变量或者单个对象结束整个生命周期后,对垃圾回收机制所占用的资源进行全部释放。运用垃圾回收机制,可以实现对空闲内存空间的最大化利用,避免内存空间浪费问题出现。目前,在现代化计算机运行性能的不断提升下,回收垃圾所耗费时间较短,此时,可以借助垃圾回收机制,频繁多次地回收程序垃圾。

4.1.2 垃圾回收方法

目前,常用的垃圾回收方法主要包含 finalize() 方法、System.gc() 方法、主 GC() 方法。接下来,对这些方法进行详细介绍。

第一, finalize() 方法。

Java 面向对象编程语言通过调用 finalize() 方法,可以达到回收程序垃圾的目的。例如:URL 地址、端口、文件等被成功打开后,自动调用垃圾收集器,实现对程序垃圾的快速化回收和处理。

第二, System.gc() 方法。

通过调用 System.gc() 方法,可以主动启动程序中的垃圾回收机制,确保垃圾回收效率和效果。System.gc() 方法具体格式如下:

```

ClassTestGC
{
    Public static void main(String[] args)
    {
        New TestGC();
        System.gc();
        System.runFinalization();
    }
}
  
```

第三,主 GC() 方法。

尽管 GC 方法对系统影响程度相对较小,但是,主 GC() 方法可以对系统影响程度较大,因此只有在特定的情况下,才能触发和调用主 GC() 方法。触发 GC() 方法的条件主要包含以下两个:首先,在没有运行应用程序情况下,应用线程长时间处于空闲状态,此时可以直接调用 GC() 方法。其次,在堆内存不充足的情况下,系统会对主 GC() 方法进行强制调用,以达到回收系统内存空间的目的。

4.2 Python 垃圾回收机制

Python 面向对象编程语言具有简洁性特征,程序开发人员在无需知道对象内存分配原理和释放原理的情况下,使用该语言,直接调用 Python 解释器,促使内存资源回收向自动化、高效化方向不断发展。目前,比较简单的 GC 算法在具体应用中,主要运用计数法,这种算法实现效率相对较低,每次回收内存,均需要修改所引用的计数,这就对系统运行性能产生不良影响。Python 解释器在未来发展中,需要运用更加高效快捷的方法,快速收集垃圾。

5 Java 内存管理

5.1 Java 内存概况

在 Java 面向对象编程语言中,通过运用 new 运算符,可以将内存直接分配给指定的对象。相关核心代码如下:

```
Objectobj=newobject();
```

计算机在运行程序时,所有对象均拥有动态化内存。运用 Java 语言内部的垃圾回收机制,可以最大化利用系统资源,同时,还能避免因程序开发人员操作失误而导致一系列故障问题。在垃圾回收机制的应用背景下,不仅可以避免内存浪费问题出现,还能简化编程复杂度,保证内存资源费分配的均匀性,降低系统崩溃出现概率,使得系统运行性能得以大幅度提高。

5.2 Java 内存结构

Java 内存主要包含以下四个组成部分:①方法区。方法区属于线程共享区域,通常情况下,该区域出现收集垃圾行为的可能性较小。如果出现内存不足,无法分配新的内存资源时, Out Of Memory Error 异常错误会被程序自动抛出。②堆。堆主要用于对进程空间资源的合理化分配,在整个垃圾回收处理中发挥出重要作用。为实现内存资源的科学化分配,可以将 Java 堆划分为若干个独立的分配缓冲区,这些

缓冲区含有多个线程,但是,运用这些缓冲区所存储的数据内容均属于对象实例。将 Java 堆划分为多个独立缓冲区的目的是实现对空间资源和垃圾的高效化分配和回收。倘若 Java 堆无法对内存进行拓展, Out Of Memory Error 异常错误会被程序自动抛出。③程序计数器。程序计数器在具体应用时,可以为单个独立线程分配相应的程序计数器,然后将程序计数器视为当前线程正在运行的指示器。在执行线程期间,程序计数器可以自动指向正在执行程序的地址。④栈。栈可以为单个线程分配相应的内存空间。在执行 Java 方法时,使用栈,可以详细化描述所调用内存整个过程。当栈分配内存空间不足时,程序会自动抛出以下两种异常错误,分别是 Stack Overflow Error 错误和 Out Of Memory Error 错误。总之,方法区和堆可以为进程分配相应空间资源,这些空间资源可以被所有线程所共享使用。栈和程序计数器可以为单个独立线程分配相应的资源。

6 结语

综上所述,论文应用静态语言和动态语言,对 Java 与 Python 不同特征进行有效区分,并在分析多继承特征时,借助接口,解决 Java 单继承存在的不足和局限性,间接性地实现 Java 多继承特征。同时,还对比和分析 Java 与 Python 两种不同垃圾回收机制,在此基础上,还探讨了 Java 内存管理模块。通过本次研究,帮助相关人员全面地了解和掌握 Java 与 Python 两种不同面向对象编程语言的基本特征,为后期编程学习打下坚实的基础。总之,Java 与 Python 两种面向对象编程语言基本特征具有重要的研究价值,值得相关学者深入地研究和探讨。

参考文献

- [1] 孙婧.使用Python建立征兵人数预测模型[J].中国战略新兴产业,2020(16):129-130.
- [2] 魏蓉.计算机软件JAVA编程特点及其技术运用研究[J].智库时代,2021(33):261-262.
- [3] 吴慧.JAVA编程在计算机应用软件中的应用特征与技术研究[J].信息技术时代,2022(12):49-51.
- [4] 刘雪花.计算机软件JAVA编程特点及其技术探究[J].科技风,2021(23):76-78.
- [5] 高源.计算机软件开发中JAVA编程语言的应用[J].网络安全技术与应用,2021(1):40-42.