20 Further Programming 进阶程序设计

20.1 Programming Paradigms 编程范式

declarative programming 声明式编程

a programming paradigm in which programs specify the desired result, using facts, rules and queries, etc., rather than how to get to it. Include logical and functional programming.

logical programming 逻辑编程

states a program as a set of logical relations. Example: Perl.

functional programming 函数式编程

constructed by applying functions to arguments using a mathematical style. Example: Lisp, Haskell.

imperative programming 命令式编程

a programming paradigm in which programs uses variables to record program state and comprises an explicit sequence of commands that update the variables, with or without procedure calls. Purely imperative programming languages include: C, Pascal, etc. Mostly imperative programming languages include: C++, Java, Python, etc.

low-level programming 低级编程

Programs using the instruction set of a processor i.e. assembly language.

object-oriented programming 面向对象编程

Programs using the concepts of class, inheritance, encapsulation and polymorphism.

class 类

a composite data type that combines a collection of data with the methods that operate on that collection.

instance 实例

a specific object based on the class.

attribute 属性

a data item defined in a class, describing a property of an instance.

method 方法

the subroutine (function or procedure) of a class acting on the attributes, implementing the behaviours of an instance.

constructor 构造器

a method that is called when an new instance is created.

getter 读取方法

a method that is used to return the value of a property.

setter 设置方法

a method that is used to update the value of a property.

encapsulation 封装

the process of putting data and methods together as a single unit, and hiding some of the data and methods from the external world.

inheritance 继承

enables the defining of a derived class that inherits from a parent class. The attributes and methods from the parent class are made available to the derived class.

polymorphism 多态

allows methods to be redefined for derived classes, so that the same method to take on different behaviours depending on which class is instantiated.

20.2 File Processing and Exception Handling 文件处理与异常处理

exception 异常

an event that occurs during the execution of a program that disrupts the normal flow of instructions, halting the program. Causes include, for example, hardware failure, programming error, user error.

exception handling 异常处理

the process of responding to an unexpected event when the program is running (such as runtime errors), so that the program does not halt unexpectedly, and produce meaningful error messages.