极客时间

零基础学 Java

1

Collection 类族简介

- 数据结构（Data Structure）
- 认识 Collection 类族

数据结构（Data Structure）

极客时间

- 数据结构是组织数据的方式，我们可以朴素的认为，数据结构 + 算法 = 程序
- 数组 （Array） 就是一种最基本的数据结构，编程语言一般本身就支持这种数据结构
- 计算机中基础的数据结构有 List，Set， Queue，Map。比较高级一点的有 Tree，Heap。这些数据结构需要代码来实现。这些实现也是一个个的类，只是专注的问题更抽象和通用
- 数据结构和算法这门课和编程语言是同等量级的课，内容非常多。在这里我们只是做了一个最基本的介绍

认识 Collection 类族

极客时间

- Collection 代表一堆元素，中文一般翻译为集合
- 看代码：Collection 是什么，了解 Collection 接口中的方法
- Collection 接口的继承者和它们的实现构成了我们所谓的 Collection 类族

极客时间

2

Collection 中的 List

Collection 中的 List

- List 代表有顺序的一组元素，中文一般翻译为链表。顺序代表遍历 List 的时候也是有顺序的
- 看代码：我们使用数组实现一个最简单的 List 接口
- 看代码：我们使用引用实现一个最简单的 List 接口
- 看源代码：常用的 Java 中对 List 的实现类
- 理解面向接口编程：接口指定规范，实现针对具体的情况做选择

3

Collection 中的 Set

Collection 中的 Set

- Set 代表一个元素不重复的集合，也就是说，Set 中的元素两两不相等
- 看例程：使用 HashSet
- 看源码：学习 Java 中 Set 的最常用的实现类 HashSet。HashSet 顾名思义，是使用了元素的 hash 值帮助做去重的
- hashCode 和 equals 符合这样一个约定：equals 返回 true， hashCode 必须相等。很多 Java 类库中的代码都是按照这种约定使用这两个方法的，比如 HashSet。这也是为什么我们要求如果一个类覆盖了 hashCode 方法，就一定要覆盖 equals 方法，并保证方法的实现符合上述约定

4

泛型简析

- 泛型的使用
- 泛型的定义

泛型的使用

- 泛型的英文名叫做 generics，一系列和泛型相关的名词都是以 generic 为前缀的，比如后面我们马上要学习的 Generic Method，Generic Types等
- 看例程：让一个 List 帮我们存储多个 String 对象
- 看例程：使用泛型，让一个 List 里只有 String
- 解决的问题：让 List 中只有一种类型的元素，使用时不用强制类型转换

泛型的定义

- 在方法中定义泛型，即 Generic Methods
- 在类型中定义泛型，即 Generic Types。类型可以是类，也可以是接口
- 看例程：定义 Generic Methods 和 Generic Types
- 看源码：ArrayList 里是怎么使用泛型的
- 解决的问题：定义泛型，让类中的代码有类型约束信息

5

再探泛型

- 有界类型
- 泛型的深水区：协变和逆变
- 泛型必须记住的两句话

有界类型

- 泛型类型不可以调用方法，因为不知道是什么类型。如果需要使用某个类的方法，则需要给定类型的范围
- 看例程：在类型定义中给定类型的范围
- 解决的问题：让自己类的代码可以调用泛型类型的方法

泛型的深水区：协变和逆变

- 协变和逆变是泛型中比较绕的点。Java 泛型对协变和逆变的支持是为了支持范围更广的参数类型
- 看例程：理解什么是协变和逆变。协变和逆变是针对引用类型而言的，可以用在返回值类型，参数类型，等引用类型上。创建对象的时候，不能使用协变和逆变
- 写入使用逆变，读取使用协变
- 解决的问题：让参数和返回值等引用类型的泛型类型更灵活

泛型必须记住的两句话

- 编译期检查并类型擦除
- 使用时的类型转换
- “教练，我想定义个泛型！” “不，你不想”

6

Iterator 接口

Iterator 接口

- 看源码：理解 Iterator 接口的意义
- Iterable 接口：实现这个接口就可以支持 forEach 循环
- 看例程：让我们的 List 使用泛型，并真正实现 Iterable 接口

7

Map：key 和 value 的映射

Map：key 和 value 的映射

极客时间

- Map 和 List 一样，是一种最基础的数据结构，它描述的是一个 key 和一个 value 一一对应的数据结构。每种高级语言都有对 Map 的定义和实现
- 看源码：Map 接口的定义
- 看源码：Map 最常用的实现类 HashMap
- 看例程：学习 Map 的使用
- 使用自己写的类作为 key， 必须保证 hashCode 和 equals 方法都实现的妥妥的，而且~~最好~~一定是不可变的。如果作为 key 的对象是可变的，多可怕

8

注解：元数据的搬运工

- 注解是什么
- 定义自己的注解

注解是什么

- 注解的英文名叫做 annotation。是给类，方法以及成员变量等元素增加元数据（metadata）的方式。换言之，就是描述这些元素的。和注释不同的是，这些描述会被 Java 编译器处理而非跳过

- 看看之前见过哪些注解，还有哪些常用的注解

定义自己的注解

极客时间

- 看例程：定义自己的注解并获取注解内容
- 通过例程可以看到，注解只是一种 metadata 传递的渠道，本身并没有实现功能
- 注解背后具体的功能，还要用代码读取注解，然后根据注解来实现相应的功能，所以每个注解的具体功能要分别学习。注解在 Spring， 测试等框架中被广泛使用

9

lambda V.S. 匿名类

- lambda V.S. 匿名类
- lambda 的理解和使用

lambda V.S. 匿名类

极客时间

- lambda 是函数式编程。很多语言中，函数（方法）是一等公民，无需依附于任何其它元素即可存在，并可以作为参数和返回值。而 Java 中只有类是一等公民，方法必须依附于某个类。 Java 现在也支持 lambda 了
- 看例程：两种方式遍历 List，遍历 Map
- 看例程：lambda 相当于是 Java 通过一顿后台操作帮我们生成了一个类来实现接口，并调用我们提供的方法
- 看例程：使用 stream 和 collector，理解 lambda 的美
- 看例程：lambda 可以有返回值和异常

lambda 的理解和使用

极客时间

- 理解 lambda 的精髓：让代码脱离类的束缚，自由的飞翔。这样就可以把代码传递给数据提供方，而不是只能把数据传递给代码。通过这种方法，达到链式的处理数据，美滴很
- Java 现在版本中 lambda 在最终的实现上其实也是使用类的。可以看到，我们在调用代码的时候，其实还是通过接口。只是 Java 帮我们把如何用我们提供的 lambda 来实现这个接口的细节隐藏掉了，而且隐藏的很好
- 限制：lambda 可以取代只有一个抽象方法的接口
- lambda 的使用：缘分到了，感觉自然了，就用就好了，缘分不到莫强求

10

基本类型的自动装箱和拆箱

基本类型的自动装箱和拆箱

- 看例程：了解数字的基本类型的封装类和常用方法
- 看例程：了解字符的基本类型的封装类和常用方法
- 看例程：了解布尔的基本类型的封装类和常用方法
- Java 是通过创建实例或者返回缓存住的实例来实现自动封箱的，是通过调用相应的转换方法实现自动拆箱的

11

Java 中的 File 类

- 理解什么是文件
- 使用 File 类操作文件和文件夹

理解什么是文件

极客时间

- 文件是操作系统对磁盘上数据的组织形式。文件包括文件路径和文件名。文件后缀其实是文件名的一部分。文件不一定要有后缀，但是一定要有文件路径和文件名，后缀是为了让操作系统更好的分辨文件的类型，以便对文件进行正确的操作

- 所有的文件，不管是什么后缀名，都是一堆在磁盘上的二进制数据。这些二进制数据需要被正确的解析，文件才能被正确的使用。比如 pptx 文件，我们也可以用文本编辑器打开它，但是文本编辑器并不能正确的解析它。

- 即使是压缩文件，其实也只是一个文件，它通过内部的组织，将很多文件的数据以及目录结构信息，压缩到一个文件的内容中

使用 File 类操作文件和文件夹

极客时间

- 使用 File 可以判断一个路径是不是文件，文件夹，是不是存在。也可以创建/重命名/删除文件夹，文件
- 看例程：学习使用 File 类进行上述操作

12

Java IO 简介

- Java 中支持的三种 IO
- Java IO 中的类和接口

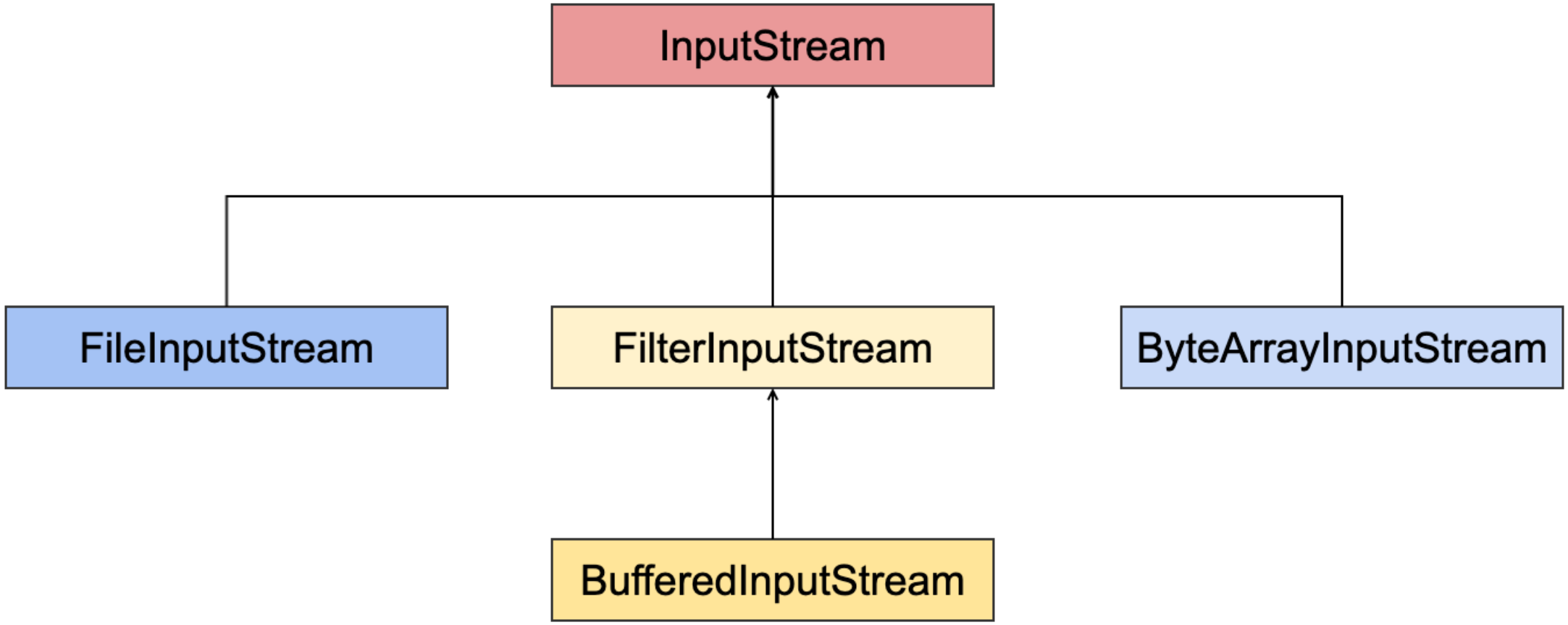
Java 中支持的三种 IO

- IO 也可以写做 I/O，是 Input / Output的缩写，也就是输入输出。这里的输入输出是指不同系统之间的数据输入输出，比如读写文件数据，读写网络数据等

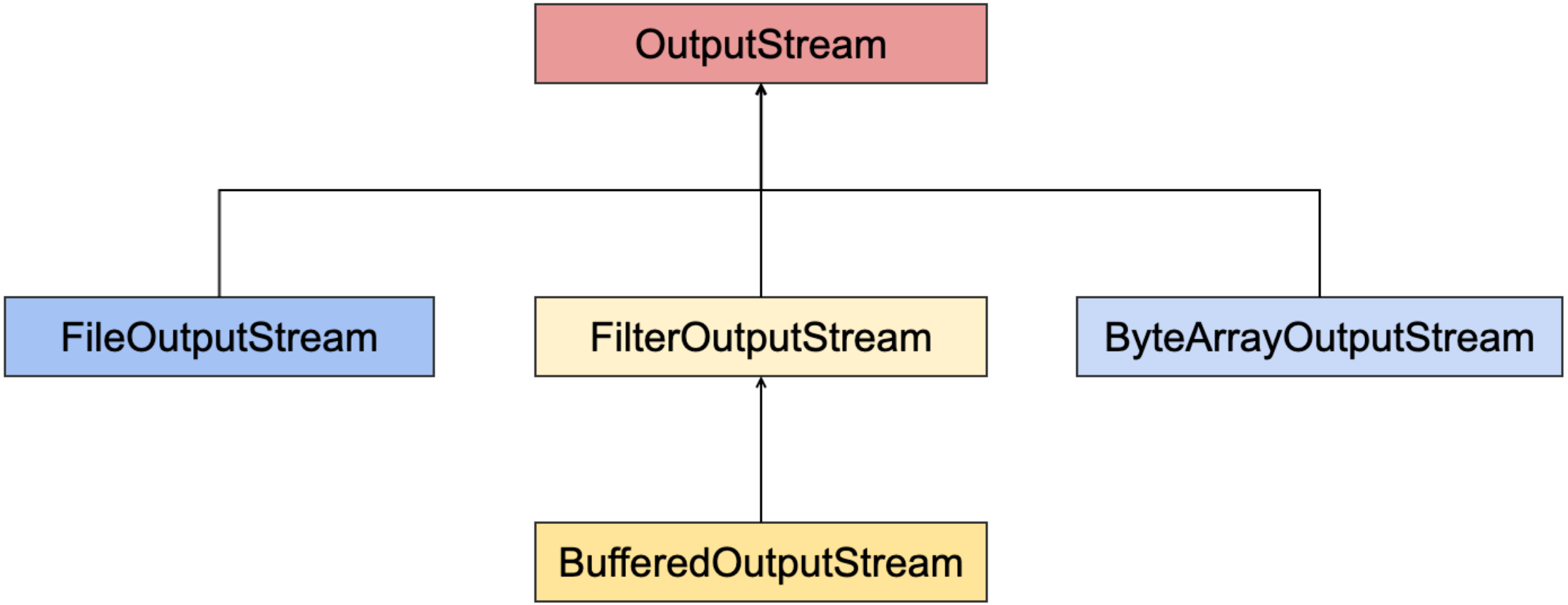
- Java 中有三代 IO 框架，分别是第一代的流式阻塞 IO （Blocking IO），第二代的 NIO （New IO）是非阻塞的，第三代 NIO2（有些地方叫做AIO，即 async IO） 又进一步支持了异步 IO

- 在这个教程中，我们学习流式阻塞 IO，这个对我们平时的学习已经足够了，也是最简单和易于理解的一种。如果是高吞吐量的生产环境下，建议使用更加优秀的 netty 作为 IO 框架

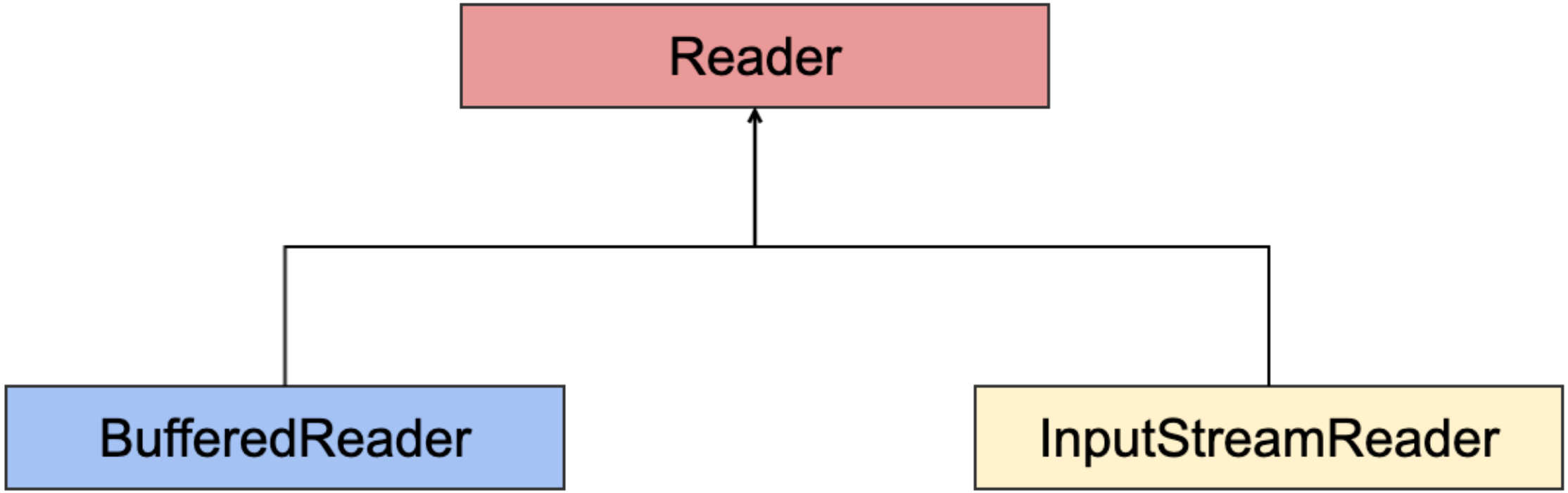
Java IO 中的类和接口：InputStream



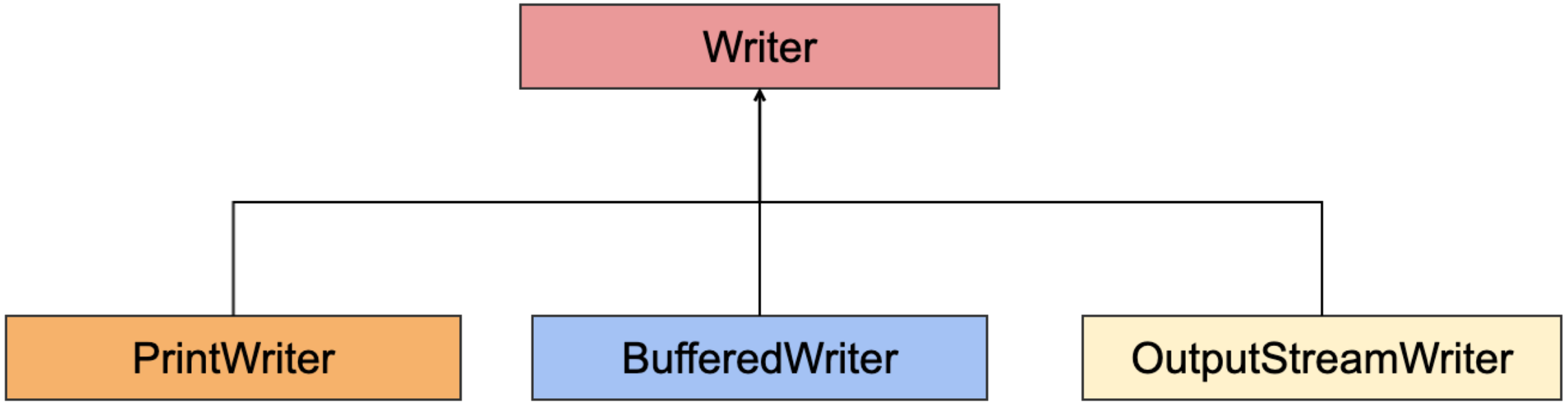
Java IO 中的类和接口：OutputStream



Java IO 中的类和接口：Reader



Java IO 中的类和接口：Writer



13

写文件内容小程序

写文件内容小程序

- 用 Scanner 读取输入，并把输入的内容写入我们指定的文件
- 看例程：学习使用套娃模式（学名装饰模式，英文名 decorate）创建出来的对象，把我们输入的文字写入到文件
- 回忆一下字符集和编码
- 会议一下 try-with-resource

极客时间

14

读文件内容小程序

读文件内容小程序

极客时间

- 看例程：学习读取指定文件的内容
- 理解 Java IO 中大量使用的装饰模式
- System.in 也是一个数据输入流
- lambda 用顺手了会很6

极客时间

15

网络通讯名词简介

网络通讯名词简介

- IP：Internet Protocol，即互联网协议。IP 现在通用的版本是 IPv4 。地址是四个 byte 的数字，用点分开。正在实施的 IPv6 地址会复杂很多。每个机器（网卡）都有一个 IP 地址。IP 地址又分内网和外网地址，就像是我们的小名和官方大名。只有外网地址才能用来外网通讯

- Port：端口。和 IP 地址一起，可以唯一的确定一个网络连接的目的地。计算机的端口是用无符号的 16 个 bit 表示的，所以端口的范围是0 ~ 65535。不能超过这个范围

网络通讯名词简介（续 1）

极客时间

- Socket：套接字，建议大家忘记这个翻译，就称呼它为 Socket 好了。Socket 就好像插座，可以服务不同的目的。我们经常说的连到什么机器的什么端口，建立这个连接就叫做建立 Socket 连接。

- Socket 包含本地的 （IP 地址 + 端口）以及远程的（IP 地址 + 端口）

网络通讯名词简介（续 2）

极客时间

- 服务器：处理客户端网络请求的机器。监听本机某个端口，等着客户端使用服务器的 IP + 端口来建立网络连接。连接建立起来之后，就可以进行数据的交换了。服务器要先启动，否则客户端连接就会失败，所以服务器有时候又被称作伺服器，长时间启动着，等着客户端的连接

- 客户端：通过指定服务器 IP 和端口，连接到服务器端

- 客户端和服务器的数据交换是独立的，可以同时进行数据的输入和输出

网络通讯名词简介（续 3）

极客时间

- 域名和 DNS（Domain Name System）：一般服务器都会使用域名让客户端连接，DNS 可以将域名翻译为 IP 地址
- 我们平时说的上网，就是通过域名，连接服务器的 80 端口，读取内容，并让浏览器帮我们去渲染，生成千姿百态的网页。80 端口是默认的 http 协议的端口
- http 协议 Hyper Text Transfer Protocol，说人话就是我们平时上网的协议

极客时间

16

简单的网络通讯小程序

简单的网络通讯小程序

极客时间

- 看例程：使用 ServerSocket 和 Socket 让本机的两个进程可以通过网络建立连接，然后你一句我一句的聊天

17

简单的抓取网页内容的程序

简单的抓取网页内容的程序

极客时间

- 看例程：使用 IP/域名和端口连接到服务器，发送请求，接收服务器端的返回内容，并把内容输出到控制台

18

JDK 和 JRE

JDK 和 JRE

- JDK 是 Java Development Kit，是 Java 的开发套件
- JRE 是 Java Runtime Environment，是运行 Java 的环境
- JDK 中包含了 JRE，而且还包含了很多和开发调试程序有关的工具
  - javap
  - jar
  - jps
  - jstack
  - jmap
  - jstat
  - jhat
  - Jvisualvm

JDK 和 JRE

- JDK 中自带的工具文档详解

Java 8
https://docs.oracle.com/javase/8/docs/technotes/tools/index.html

Java 11：
https://docs.oracle.com/en/java/javase/11/tools/tools-and-command-reference.html