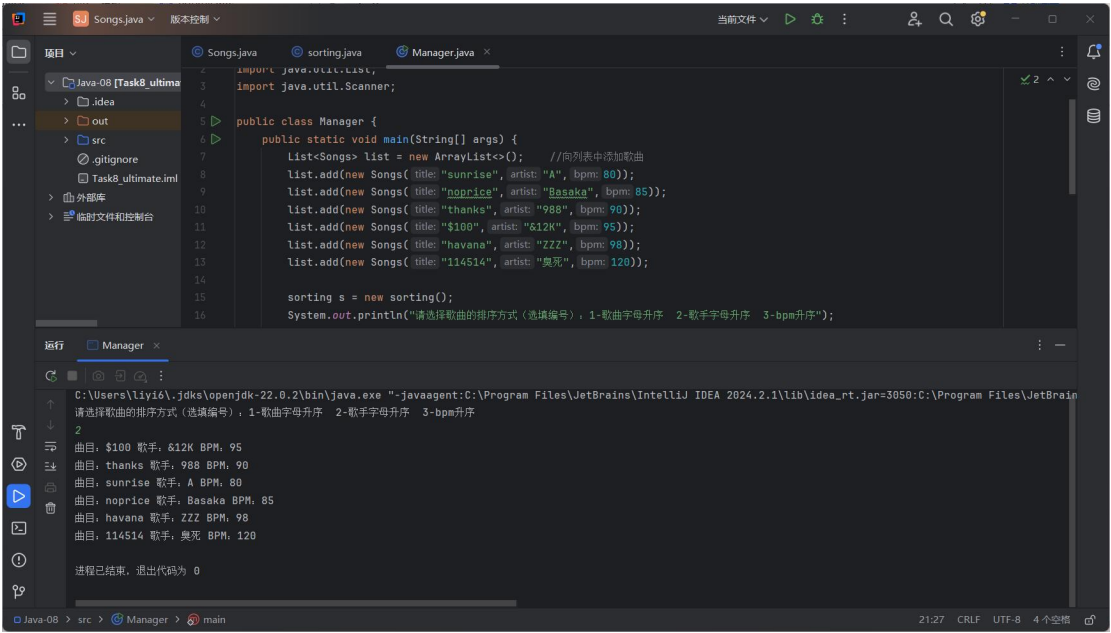
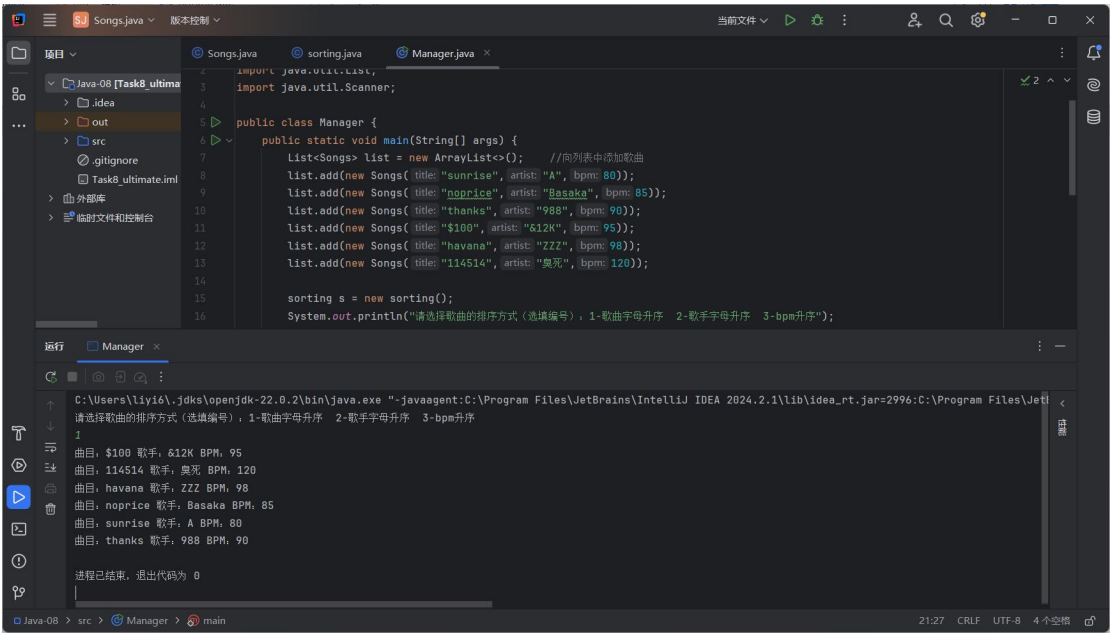
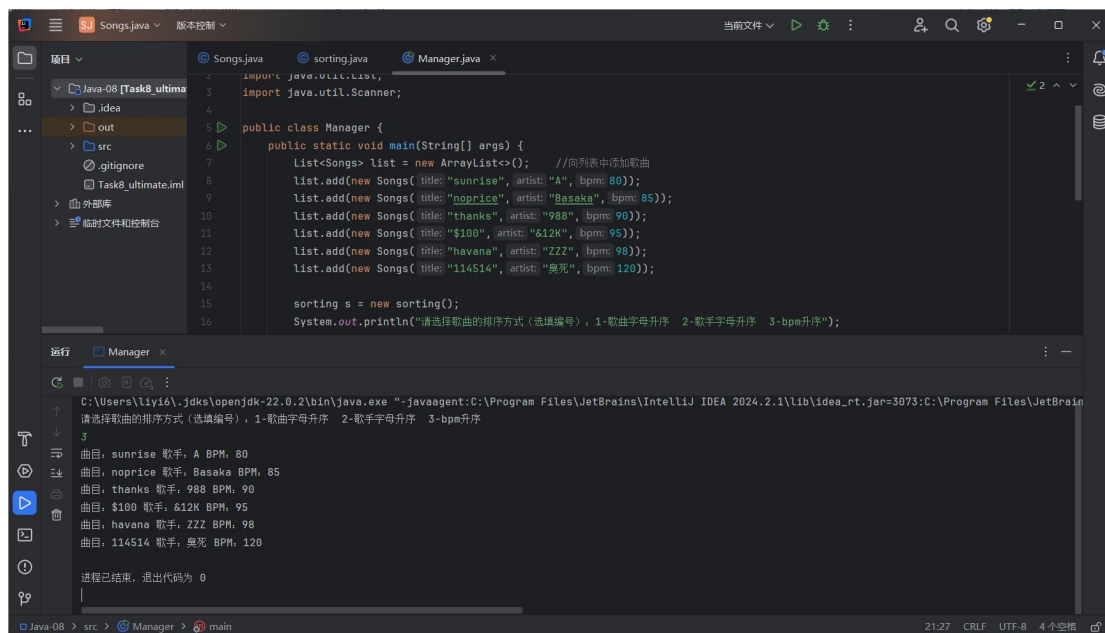


三种排序方式的运行结果截图





说明：（项目源文件已上传至两个文件夹）

本题通过两种方式实现了添加歌曲，并通过歌曲中的三个属性进行排序，其中通过泛型方法的项目比较方便的实现了该功能。泛型方法让我能在同一个类中储存不同类型的数据，同时因为它本身把类型作为一个参数，故在编写的过程中始终有类型安全，无需每次进行类型转换。但在用泛型的过程中，原来程序的排序功能出现了报错，经查资料发现是由于 `Comparator.comparing` 方法需要类型是可以被比较的，而泛型中的 `T` 能代表任何类型，故该方法无法判断类型是否可以被比较，故在 `songs` 类改为了 `class Songs<T extends Comparable<T>>`。让它被限制为实现了 `Comparable` 接口的类型，使 `T` 能被比较。然后在 `ArrayList` 中泛型参数 `songs` 限定为了 `<String>`，明确了类型使其能被比较。

最后，排序代码由判断改为了 `switch-case` 语句，使其能够更快运行。