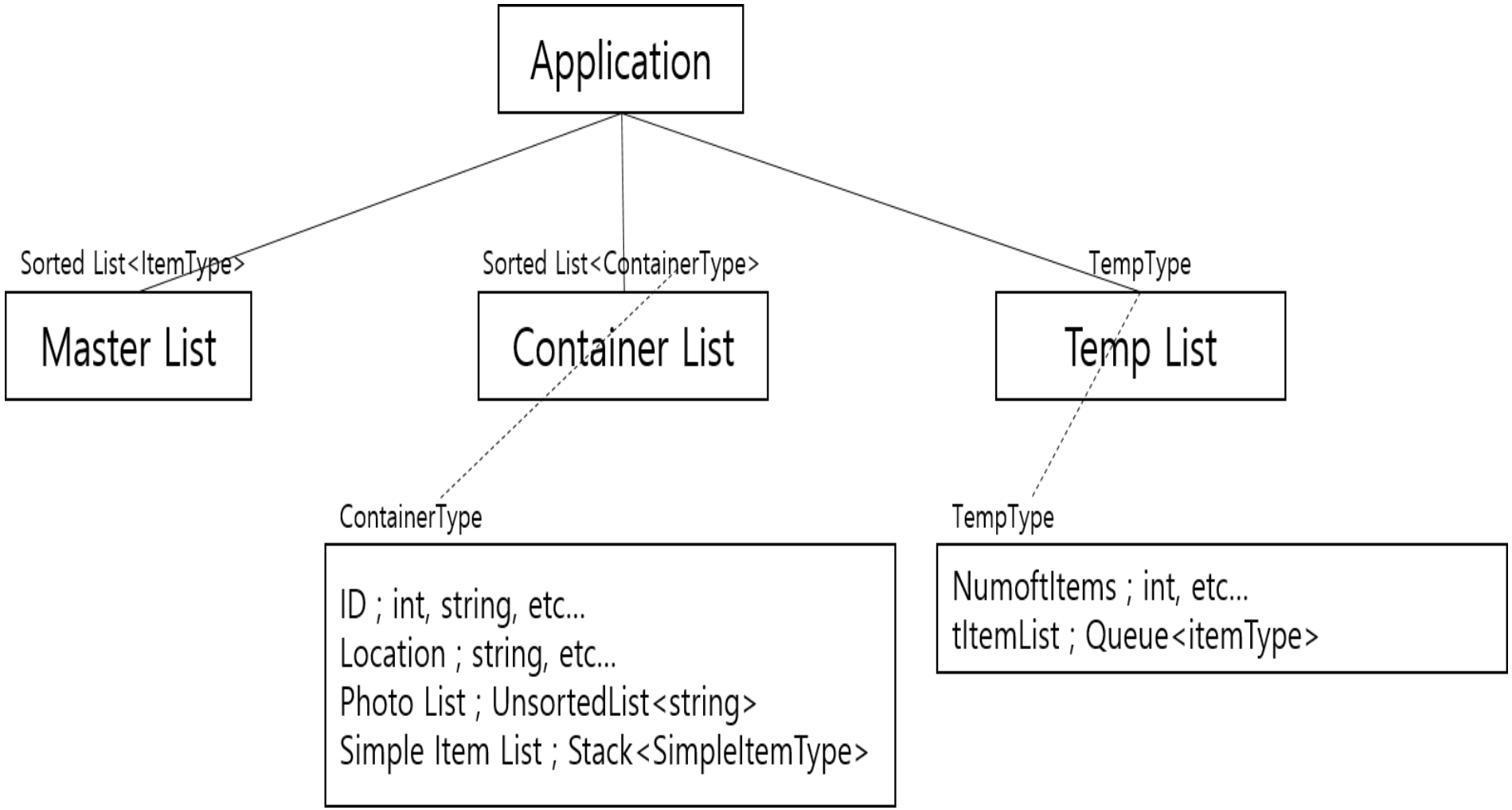
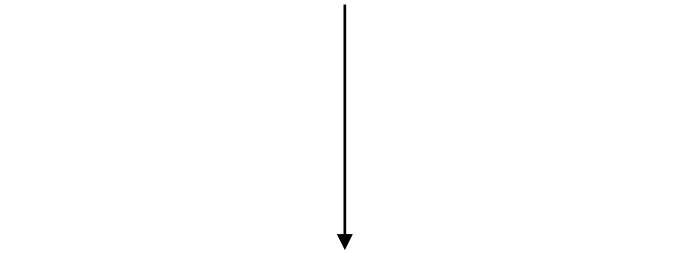

Application
Sorted List<ItemType>
Sorted List<ContainerType>
TempType
Master List
Container List
Temp List
ContainerType
TempType
ID ; int, string, etc...
Location ; string, etc...
Photo List ; UnsortedList<string>
Simple Item List ; Stack<SimpleItemType>
NumoftItems ; int, etc...
tItemList ; Queue<itemType>

기존 ADT

Base

static SortedList<ItemType>
static SortedList<ContainerType>
static TempType



Application : public Base

```h
#pragma once

class ContainerType;
class ItemType;
class SimpleItemType;
class TempType;

template <typename T>
class SortedList;

class Base
{
public:
    static SortedList<ItemType>     MasterList;     ///< master list.
    static SortedList<ContainerType>    ContainerList;  ///< constainer list.
    static TempType                 TempList;       ///< temp item list.
};
```

List들을 모아놓는 class를 새로 만듦.
Static으로 class를 선언하여 상속된 모든 객체들이 같은 memory를 공유.

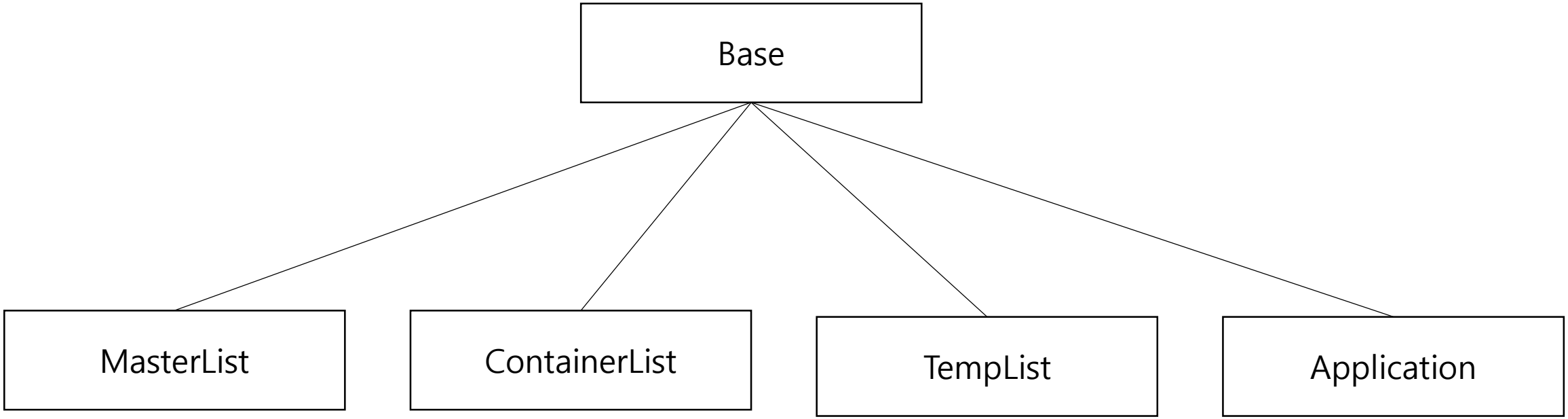
```h
class Application : public Base
{
private:
    ifstream                    m_InFile;       ///< input file descriptor.
    ofstream                    m_OutFile;      ///< output file descriptor.
    int                         m_Command;      ///< current command number.

    /*SortedList<ItemType>      MasterList;     ///< master list.
    SortedList<ContainerType>   ContainerList;  ///< constainer list.
    TempType                    TempList;       ///< temp item list.*/
```

```
class ContainerType : public Base
{
```

```
86
87  void ContainerType::DisplayAllDetailsItem()
88  {
89      SimpleItemType tmpItem;
90      sItemList.ResetList();
91      while(sItemList.GetNextItem(tmpItem) != -1) {
92          ItemType detail;
93          detail.SetId(tmpItem.GetId());
94          MasterList.Get(detail);
95          cout << detail;
96      }
97  }
98
99  void ContainerType::DisplayDetailItem(SimpleItemType Item)
100 {
101     ItemType detail;
102     detail.SetId(Item.GetId());
103     MasterList.Get(detail);
104     cout << detail;
105 }
```

Base class를 상속하면,
원래의 하위 계층에서도
다른 list에 접근 및 수정이 가능함.

SortedList<ItemType> MasterList	SortedList<SimpleType> ContainerList	TempType TempList



- 각 List는 처음 Base를 생성할 때 한 번만 생성된다.
- 어떠한 class에서 접근하던지 똑같은 공간(Base)에서 처리된다.
- 어디서 수정하던지 모든 곳에서 값의 변경이 이루어진다.
- 하위 계층에서 상위 계층의 값을 다룰 때 유용하다.