

## 作业要求

完成 `MyList.h` 中规定的函数功能，完善 `IntList` 类，使其适用多种数据类型。`MyList.h` 不做任何改动，只实现对应 `MyList.cpp` 中的功能，提供 `main` 和 `result` 可做测试，注意不要照抄 PPT 中的代码，需要完善代码，注意空指针等各种边界条件处理、注意检测越界并输出越界信息。

### MyList 类功能

管理一个可动态扩展的列表，并进行运算符重载。

1 支持异常检测 --在[]中检测越界，在调用[]的地方进行处理，如出现越界，抛出 `My_Out_of_Range_Exception` 类型异常

2 改进 `IntList` 的内存管理方法

-- 增加私有 `capacity` 数据域，表示当前 `list` 中可用于存储的 `list` 长度，`numberValues` 表示已经存储的 `list` 长度。`capacity` 的最小初值为 4，`capacity` 的值应保证大于等于 `numberValues`。

-- 增加公有 `getCapacity()` 函数，返回当前 `capacity` 数据值。

-- 改进涉及内存管理的函数，如构造函数、`resize` 函数等。当 `numberValues` 需要大于 `capacity` 时，要先将 `capacity` 翻倍若干次以保证 `capacity` 大于或等于 `numberValues`，所以 `capacity` 的值以 4, 8, 16, 32, 64....的规则增加。若存储空间足够的时候不需要做内存操作。

-- 增加公有 `void reserve(int newCapacity)` 函数，用于为 `list` 分配指定长度的内存。当 `newCapacity` 小于等于成员变量 `capacity` 时，不做任何操作。当 `newCapacity` 大于成员变量 `capacity` 时，扩展列表容量至 `newCapacity` 的值，即使 `newCapacity` 的值并不是 2 的整数次幂。

3 改进 `IntList` 为 `MyList`，支持模版

--改进已有函数，支持模版

--支持 `Number` 系列类，`int`，`double`，`float`，`char`，`string` 类型的使用

### MyList.h 解释

成员数据为私有，`numberValues` 为列表内已有数据的长度，`capacity` 为列表可容纳数据的长度，`values` 为指向列表数据的指针。

成员函数为公有，包括如下函数：

// 构造函数，第一个参数为列表长度，第二个参数为列表中数据的初始值

`MyList(int numVals = 0, Tval = T());`

// 复制构造函数，使当前对象与il对象相同

`MyList (const T&il);`

// 析构函数，对象退出前的内存清理

`~ MyList ();`

// 等于运算符，使当前对象与il对象相同

`T& operator=(const T&il);`

```

// 返回列表长度
int size() const;
// 返回容量长度
int getCapacity() const;
// 改变列表长度
// 当n大于等于原来的size，使用val初始化多出的元素
// 当n小于原来的size，直接截断原列表
void resize(int n = 0, T val = T());
// 改变列表容量，当newCapacity小于等于成员变量capacity时，不做任何操作。
// 当newCapacity大于成员变量capacity时，扩展列表容量至newCapacity的值。即使
newCapacity的值并不是2的整数次幂。
void reserve(int newCapacity);
// 将val插入到当前列表的最后
void push_back(T val);
// 将列表转换成字符串
std::string toString() const;
// 打印出列表内容，已给出实现
void print() const//print the list
//取出index所在位置的整形数，越界需要抛出指定异常
T& operator [] ( const int index);
//取出index所在位置的整形数，越界需要抛出指定异常
const T& operator [] ( const int index) const;
//判断参数中的两个IntList是否相等，相等返回true，返回false
//相等定义为list长度和list中的每个数据都相等
bool operator == (const T& il1, const T& il2);
//判断参数中的两个IntList是否相等，相等返回false，不等返回true
//相等定义为list长度和list中的每个数据都相等
bool operator != (const T& il1, const T& il2);
//参数中的两个IntList相加
//两个list不等长的情况下，短list中差的元素视为0
IntList operator + (const T& il1, const T& il2);
//参数中的两个IntList相减，il1-il2
//两个list不等长的情况下，短list中差的元素视为0
IntList operator - (const T& il1, const T& il2);
//输出il，支持文件输出，内容直接输出toString()的返回值，非友元函数
std::ostream& operator << (std::ostream& ost, const T& il);
//由用户输入il的内容，支持文件输入，非友元函数
//输入格式：输入list长度，依次输入list的值，每个数字以空格或者回车间隔，输入完
成以回车结束
std::istream& operator >> (std::istream& ist, T& il);

```