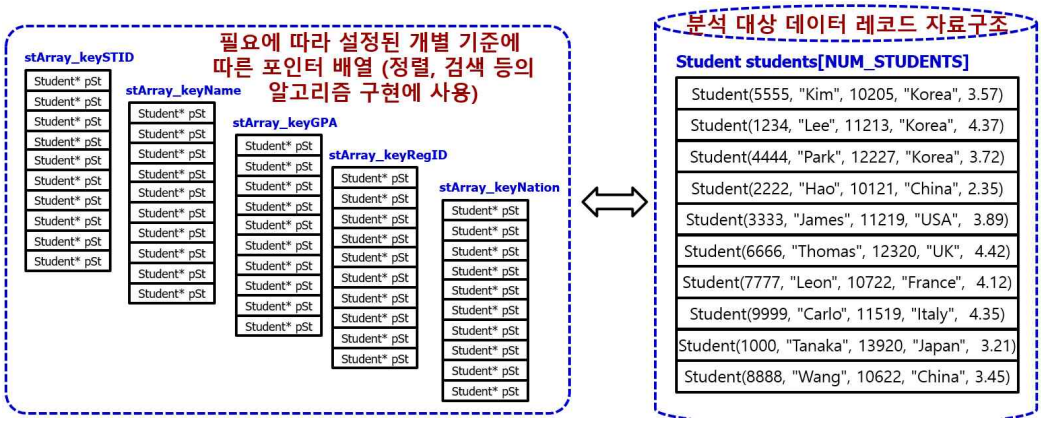
2022-2 객체지향형 프로그래밍과 자료구조 Exam2C

2022. 10. 21.
담당교수 김영탁

2C. class Person, class Student, Template class T_Array<T, K>

2C.1 기본 개념



2C.2 class Person

```
/* Person.h */
. . . . // 필요한 헤더파일 내용 추가
class Person
{
    friend ostream& operator<< (ostream& fout, const Person& p)
    {
        . . . . // 직접 구현
    }
public:
    Person() { name = "nobody"; }
    Person(string n) { name = n; }
    . . . . // 필요한 추가 멤버함수가 있는 경우 추가 구현
protected:
    string name;
    int regID;   // 주민등록번호
    string nation;   // 국가
};
```

- class Person는 데이터 멤버로 이름 (name), 주민등록번호(regID), 국가(nation)을 가짐
- class Person 관련 멤버함수들은 Person.h에 inline 함수로 구현 또는 Person.cpp 파일에 포함시킬 것

2C.3 class Student

```
/* Student.h */
. . . . // include necessary header files and definition
class Student : public Person
{
    friend ostream & operator<< (ostream &, Student &);
public:
    Student();  // default constructor
    Student(int s_id, string n, Date dob, Time avt, double gpa);
    void getKey(string keyName, void* pKey);
    . . . . // 필요한 추가 멤버함수가 있는 경우 추가 구현
private:
    int st_id;
    double gpa;
};
```

- class Student는 class Person으로부터 상속 받음
- class Student 및 관련 함수들은 Student.h에 inline 함수로 구현 또는 Student.cpp 파일에 작성할 것

2C.4 class T_Array<T, K>

```
template<typename T, typename K>
class T_Array
{
public:
	T_Array(int n, string nm); // constructor
	~T_Array(); // destructor
	int size() { return num_elements; }
	bool empty() { return num_elements == 0; }
	string get_name() { return name; }
	void insert(int i, T element);
	void selection_sort(string keyName, SortingOrder sortOrder);
	void merge_sort(string keyName, SortingOrder sd);
	void print(int elements_per_line);
	bool isValidIndex(int i);
	T& operator[](int index) { return t_array[index]; }
private:
	void _mergeSort(T* arr, T* tmp_array, int left, int right, string keyName, SortingOrder sortOrder);
	T *t_array;
	int num_elements;
	int capacity;
	string name;
};
```

- class T_Array<T, K> 정의 및 멤버함수는 모두 T_Array.h 파일에 작성할 것

2C.5 class T_Array<T, K> 기능 시험용 main() 함수 구현

```
. . .  // include necessary header files and definitions
#define NUM_STUDENTS 10
void main()
{
	Student students[NUM_STUDENTS] =
	{
		Student(5555, "Kim", 10205, "Korea", 3.57),
		Student(1234, "Lee", 11213, "Korea",   4.37),
		Student(4444, "Park", 12227, "Korea", 3.72),
		Student(2222, "Hao", 10121, "China", 2.35),
		Student(3333, "James", 11219, "USA",   3.89),
		Student(6666, "Thomas", 12320, "UK",   4.42),
		Student(7777, "Leon", 10722, "France",   4.12),
		Student(9999, "Carlo", 11519, "Italy",   4.35),
		Student(1000, "Tanaka", 13920, "Japan",   3.21),
		Student(8888, "Wang", 10622, "China", 3.45),
	};

	Student* pSt;
	T_Array<Student*, int> stArray_keySTID(NUM_STUDENTS, "T_Array<Student, keyST_ID>");
	for (int i = 0; i < NUM_STUDENTS; i++)
	{
		stArray_keySTID.insert(i, &students[i]);
	}
	cout << "T_Array<Student_keyID> at initialization :" << endl;
	stArray_keySTID.print(1);

	stArray_keySTID.selection_sort(string("ST_ID"), INCREASING);
	cout << "\nT_Array<Student_keyID> after selection sorting (increasing order) by ST_ID : " << endl;
	stArray_keySTID.print(1);

	T_Array<Student*, string> stArray_keyName(NUM_STUDENTS, "T_Array<Student, keyName>");
	for (int i = 0; i < NUM_STUDENTS; i++)
	{
		stArray_keyName.insert(i, &students[i]);
	}
	stArray_keyName.selection_sort(string("NAME"), INCREASING);
	cout << "T_Array<Student, keyName> after selection_sorting (increasing order) by Name : " << endl;
	stArray_keyName.print(1);
```

```
	T_Array<Student*, double> stArray_keyGPA(NUM_STUDENTS, "T_Array<Student, keyGPA>");
	for (int i = 0; i < NUM_STUDENTS; i++)
	{
		stArray_keyGPA.insert(i, &students[i]);
	}
	stArray_keyGPA.selection_sort(string("GPA"), DECREASING);
	cout << "T_Array<Student, keyGPA> after selection_sorting (decreasing order) by GPA : " << endl;
	stArray_keyGPA.print(1);

	T_Array<Student*, int> stArray_keyRegID(NUM_STUDENTS, "T_Array<Student, keyRegID>");
	for (int i = 0; i < NUM_STUDENTS; i++)
	{
		stArray_keyRegID.insert(i, &students[i]);
	}
	stArray_keyRegID.merge_sort(string("REGID"), INCREASING);
	cout << "\nstArray_keyRegID after merge sorting (increasing order) by regID : " << endl;
	stArray_keyRegID.print(1);

	T_Array<Student*, string> stArray_keyNation(NUM_STUDENTS, "T_Array<Student, keyNation>");
	for (int i = 0; i < NUM_STUDENTS; i++)
	{
		stArray_keyNation.insert(i, &students[i]);
	}
	stArray_keyNation.merge_sort(string("NATION"), INCREASING);
	cout << "\nstArray_keyNation after merge sorting (increasing order) by nation : " << endl;
	stArray_keyNation.print(1);

}
```

2C.6 T_Array<T, K>의 기능 시험 결과

```
T_Array<Student_keyID> at initialization :
Student [ st_id:  5555, name: Kim    , gpa: 3.57 , regID:  10205, nation: Korea  ]
Student [ st_id:  1234, name: Lee    , gpa: 4.37 , regID:  11213, nation: Korea  ]
Student [ st_id:  4444, name: Park   , gpa: 3.72 , regID:  12227, nation: Korea  ]
Student [ st_id:  2222, name: Hao    , gpa: 2.35 , regID:  10121, nation: China  ]
Student [ st_id:  3333, name: James  , gpa: 3.89 , regID:  11219, nation: USA    ]
Student [ st_id:  6666, name: Thomas , gpa: 4.42 , regID:  12320, nation: UK     ]
Student [ st_id:  7777, name: Leon   , gpa: 4.12 , regID:  10722, nation: France ]
Student [ st_id:  9999, name: Carlo  , gpa: 4.35 , regID:  11519, nation: Italy  ]
Student [ st_id:  1000, name: Tanaka , gpa: 3.21 , regID:  13920, nation: Japan  ]
Student [ st_id:  8888, name: Wang   , gpa: 3.45 , regID:  10622, nation: China  ]


T_Array<Student_keyID> after selection sorting (increasing order) by ST_ID :
Student [ st_id:  1000, name: Tanaka , gpa: 3.21 , regID:  13920, nation: Japan  ]
Student [ st_id:  1234, name: Lee    , gpa: 4.37 , regID:  11213, nation: Korea  ]
Student [ st_id:  2222, name: Hao    , gpa: 2.35 , regID:  10121, nation: China  ]
Student [ st_id:  3333, name: James  , gpa: 3.89 , regID:  11219, nation: USA    ]
Student [ st_id:  4444, name: Park   , gpa: 3.72 , regID:  12227, nation: Korea  ]
Student [ st_id:  5555, name: Kim    , gpa: 3.57 , regID:  10205, nation: Korea  ]
Student [ st_id:  6666, name: Thomas , gpa: 4.42 , regID:  12320, nation: UK     ]
Student [ st_id:  7777, name: Leon   , gpa: 4.12 , regID:  10722, nation: France ]
Student [ st_id:  8888, name: Wang   , gpa: 3.45 , regID:  10622, nation: China  ]
Student [ st_id:  9999, name: Carlo  , gpa: 4.35 , regID:  11519, nation: Italy  ]

T_Array<Student, keyName> after selection_sorting (increasing order) by Name :
Student [ st_id:  9999, name: Carlo  , gpa: 4.35 , regID:  11519, nation: Italy  ]
Student [ st_id:  2222, name: Hao    , gpa: 2.35 , regID:  10121, nation: China  ]
Student [ st_id:  3333, name: James  , gpa: 3.89 , regID:  11219, nation: USA    ]
Student [ st_id:  5555, name: Kim    , gpa: 3.57 , regID:  10205, nation: Korea  ]
Student [ st_id:  1234, name: Lee    , gpa: 4.37 , regID:  11213, nation: Korea  ]
Student [ st_id:  7777, name: Leon   , gpa: 4.12 , regID:  10722, nation: France ]
Student [ st_id:  4444, name: Park   , gpa: 3.72 , regID:  12227, nation: Korea  ]
Student [ st_id:  1000, name: Tanaka , gpa: 3.21 , regID:  13920, nation: Japan  ]
Student [ st_id:  6666, name: Thomas , gpa: 4.42 , regID:  12320, nation: UK     ]
Student [ st_id:  8888, name: Wang   , gpa: 3.45 , regID:  10622, nation: China  ]

T_Array<Student, keyGPA> after selection_sorting (decreasing order) by GPA :
Student [ st_id:  6666, name: Thomas , gpa: 4.42 , regID:  12320, nation: UK     ]
Student [ st_id:  1234, name: Lee    , gpa: 4.37 , regID:  11213, nation: Korea  ]
Student [ st_id:  9999, name: Carlo  , gpa: 4.35 , regID:  11519, nation: Italy  ]
Student [ st_id:  7777, name: Leon   , gpa: 4.12 , regID:  10722, nation: France ]
Student [ st_id:  3333, name: James  , gpa: 3.89 , regID:  11219, nation: USA    ]
Student [ st_id:  4444, name: Park   , gpa: 3.72 , regID:  12227, nation: Korea  ]
Student [ st_id:  5555, name: Kim    , gpa: 3.57 , regID:  10205, nation: Korea  ]
Student [ st_id:  8888, name: Wang   , gpa: 3.45 , regID:  10622, nation: China  ]
Student [ st_id:  1000, name: Tanaka , gpa: 3.21 , regID:  13920, nation: Japan  ]
Student [ st_id:  2222, name: Hao    , gpa: 2.35 , regID:  10121, nation: China  ]
```

```
stArray_keyRegID after merge sorting (increasing order) by regID :
Student [ st_id:  2222, name: Hao    , gpa: 2.35 , regID:  10121, nation: China  ]
Student [ st_id:  5555, name: Kim    , gpa: 3.57 , regID:  10205, nation: Korea  ]
Student [ st_id:  8888, name: Wang   , gpa: 3.45 , regID:  10622, nation: China  ]
Student [ st_id:  7777, name: Leon   , gpa: 4.12 , regID:  10722, nation: France ]
Student [ st_id:  1234, name: Lee    , gpa: 4.37 , regID:  11213, nation: Korea  ]
Student [ st_id:  3333, name: James  , gpa: 3.89 , regID:  11219, nation: USA    ]
Student [ st_id:  9999, name: Carlo  , gpa: 4.35 , regID:  11519, nation: Italy  ]
Student [ st_id:  4444, name: Park   , gpa: 3.72 , regID:  12227, nation: Korea  ]
Student [ st_id:  6666, name: Thomas , gpa: 4.42 , regID:  12320, nation: UK     ]
Student [ st_id:  1000, name: Tanaka , gpa: 3.21 , regID:  13920, nation: Japan  ]


stArray_keyNation after merge sorting (increasing order) by nation :
Student [ st_id:  8888, name: Wang   , gpa: 3.45 , regID:  10622, nation: China  ]
Student [ st_id:  2222, name: Hao    , gpa: 2.35 , regID:  10121, nation: China  ]
Student [ st_id:  7777, name: Leon   , gpa: 4.12 , regID:  10722, nation: France ]
Student [ st_id:  9999, name: Carlo  , gpa: 4.35 , regID:  11519, nation: Italy  ]
Student [ st_id:  1000, name: Tanaka , gpa: 3.21 , regID:  13920, nation: Japan  ]
Student [ st_id:  4444, name: Park   , gpa: 3.72 , regID:  12227, nation: Korea  ]
Student [ st_id:  5555, name: Kim    , gpa: 3.57 , regID:  10205, nation: Korea  ]
Student [ st_id:  1234, name: Lee    , gpa: 4.37 , regID:  11213, nation: Korea  ]
Student [ st_id:  6666, name: Thomas , gpa: 4.42 , regID:  12320, nation: UK     ]
Student [ st_id:  3333, name: James  , gpa: 3.89 , regID:  11219, nation: USA    ]
```

2C.7 결과물 제출

- 바탕화면에 Exam2 폴더내에 Exam2C_학번_이름 프로젝트를 생성
- 압축 파일 내에 포함사항 : 작성한 프로젝트, 실행결과 Capture(채점 시 정확한 실행 유무를 확인하기 위함)
- 실행 화면 캡쳐파일은 각 시험 섹션별로 프로젝트 폴더 내에 저장 후 시험 섹션별 폴더별로 압축
- 제출시 .vs 폴더는 삭제 후 문제별 폴더를 압축하여 제출