**게임기초프로그래밍**

**과 목 명 : 게임기초프로그래밍**

**능력단위 : 게임기초프로그래밍(C++ 프로그래밍)**

**제출일자 : 2021년 12월 16일**

**포트폴리오 : 성적 관리 프로그램 구현(LinkedList, Templete)**

**작 성 자 : 곽근오**

**<제출내역>**

1. **성적 관리 프로그램 소개**
2. **프로그램 설계 (Sequence Diagram)**
3. **분석 및 세부 문서**
4. **추가 보완 수정 사항**

**1. 성적 관리 프로그램 소개**

LinkedList를 기반으로 제작하여, 학생의 수에 따라 Node를 생성하고 정보를 추가,

학생수(NodeCount) || 학생정보(NodeInfo)

과목과 성적은 필요한 개수를 받아, 과목과 성적을 적고, STL map을 이용해 저장한다.

과목수(SubjectCOunt) || 과목(Subject) || 성적(iRecord)

텍스트이(가) 표시된 사진

자동 생성된 설명

[1-1] 학생 정보 입력 부분, AddList 함수

예외처리 부분에서, 제작단계에서 지속적으로 QA를 진행하여 메모리 오류와 Loop 방지, 필요한 기능 구현.

텍스트이(가) 표시된 사진

자동 생성된 설명

[1-2] 학생 정보 입력 부분, AddList 함수

Class Templete을 구현하여, 다양한 Node-Info형에 대응하게 제작하여, 재사용성을 높임.

(Node와 Info 규격에 맞는 다른 관리 프로그램으로 변환 가능) 텍스트이(가) 표시된 사진

자동 생성된 설명

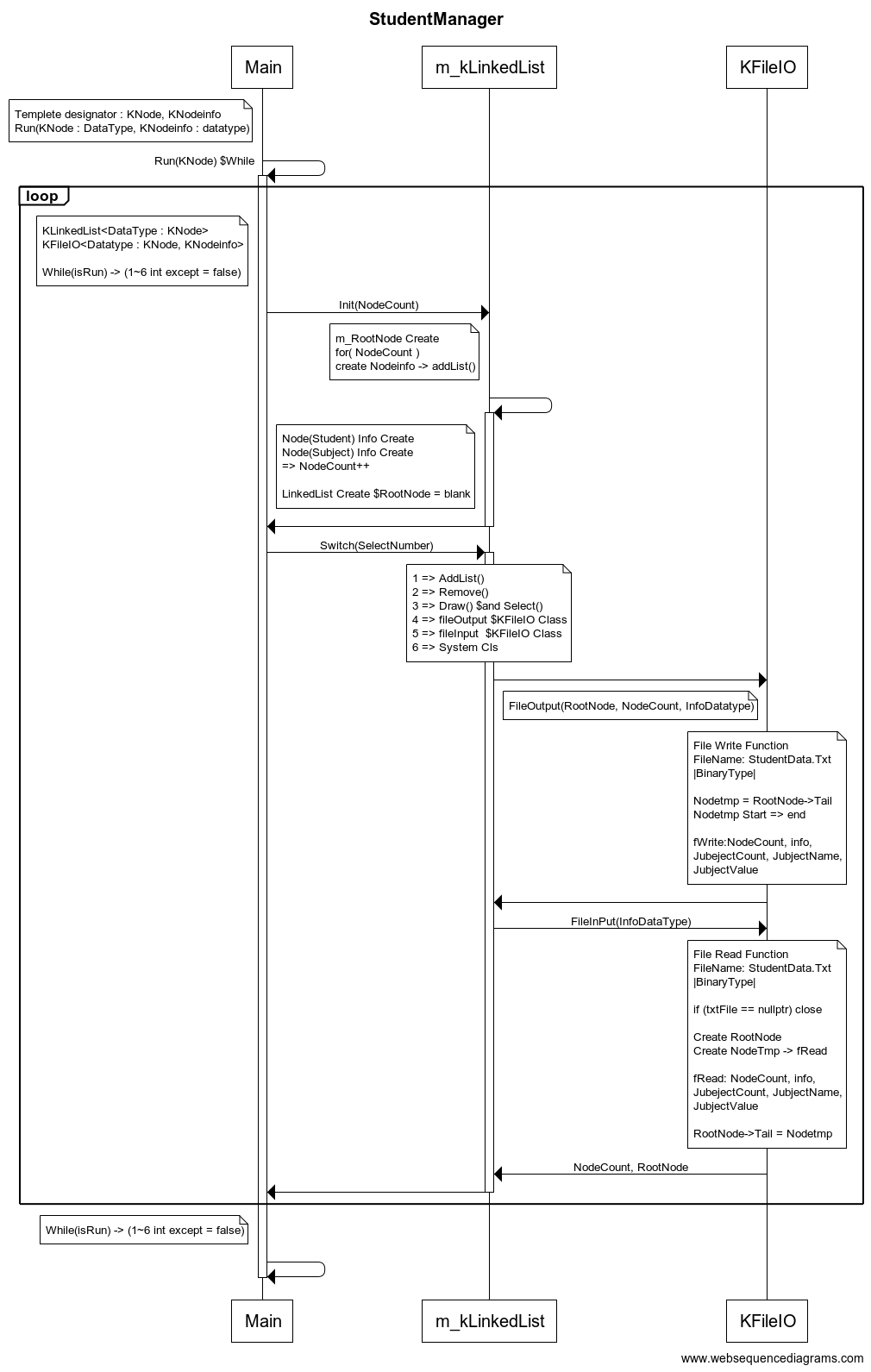
[1-3] 파일 입출력 부분, FileOutput 함수

**2. 프로그램 설계 (Sequence Diagram)**

Init() 초기화 후, LinkedList형태로 RootNode 생성 후, Node를 추가할 때마다 값과 정보를 받은 뒤 연결시킨다.

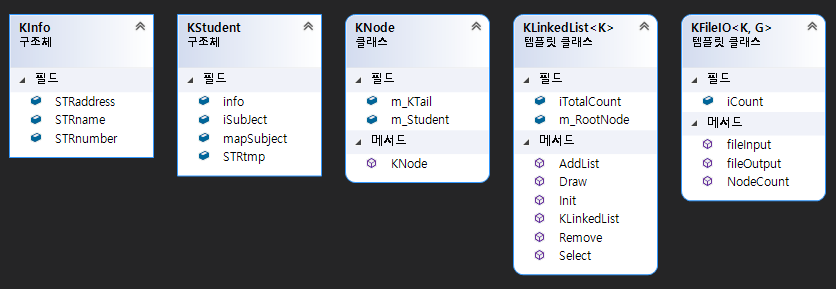
연결된 Node들은 각각의 정보를 가지고 RootNode-Tail부터 차례대로 물려있으며,

마지막 Node는 RootNode의 주소값을 가지고 있는다.



[2-1] 시퀸스 다이어그램

**3. 프로젝트 분석 및 세부 문서**



[3-1] 성적 관리 프로그램 메인 클래스

* 함수에 들어가는 인자 값은 이후 함수 문서에서 작성

1. KInfo 클래스에서 정보 데이터 (학생 이름, 주소, 전화번호)구조체 생성
2. KStudent 클래스에서 정보 데이터와 성적(mapSubject) 데이터를 합쳐,

최종 데이터 구조체 생성

1. KNode 클래스에서 LinkedList용 Node(Tail\* Node) 및 데이터 구조체 변수생성
2. KLinkedList에서 메인 LinkedList 시스템 생성, 관리, 검색, 삭제 구현
3. KFileIO 클래스를 만들어 파일 입출력 시스템을 메인 시스템에서 분리

텍스트이(가) 표시된 사진

자동 생성된 설명

[3-2] LinkedList 클래스, 메인 함수 부분

1. 클래스에서 Node 형태에(Student, College 등) 따라 변환 가능하게 클래스 템플릿화

|  |  |
| --- | --- |
| **함수** | **비고** |
| Init(int ivalue) | RootNode 생성, ivalue의 값에  따라 메인 Node 생성 |
| AddList() | 생성된 Node에 정보 기입 함수  (Init 함수에서 호출) |
| Remove(std::string NameValue) | (기본값: 전체삭제) std::string 값을 받아, 해당하는 string(이름)에 맞는 정보를 찾아 삭제 |
| Draw() | Iterator를 이용해 학생정보와 성적정볼 출력 |
| Select() | Std::stirring 값을 입력받아, 해당하는 이름에 맞는 데이터를 map iterator로 돌려서 찾은다음 출력 후 삭제 여부를 질문 후삭제한다. |

텍스트, 스크린샷, 화면, 은색이(가) 표시된 사진

자동 생성된 설명

[3-3] KFileIO 클래스, Inout 함수 부분

1. DataType 과 Node를 전달 받아, 바이너리 파일로 저장하고, 저장여부를 출력한다.
2. 저장된 파일의 DataType을 전달받아, 해당하는 Node를 불러오고 출력한다.
3. Templete을 사용하여 다양한 Node 및 Data에 재사용성을 높임.

|  |  |
| --- | --- |
| **함수** | **비고** |
| fileInput(G NodeInfoData) | 파일 읽어오기 함수, 템플릿을 사용해, DataType을 전달받아 해당하는 Node를 읽어온다 |
| fileOutput(const K\* \_Node, const int \_ivalue, G \_NodeInfoData) | 파일 저장 함수, 저장하는 용도로 사용하기 위해 const를 붙여 인자를 전달받고, 전달받은 DataType으로 Node 저장 |
| NodeCount() | 총 Node의 개수를 구하기 위한 함수 |

텍스트이(가) 표시된 사진

자동 생성된 설명

[3-4] Node 안 내부 데이터 정보

1. Info에서 메인으로 사용되는 정보(이름, 직업, 주소 등)를 입력받는다.
2. 성적 및 수치화에 필요한 정보는 Multimap을 사용해 몇 개의 데이터를 전달 받고 저장할지 준비한다.

**4. 추가 보완 수정 사항**

1. 기능의 구현은 문제 없이 진행되었으나, 데이터를 입력받고 저장하는데 있어서 최적화된 코드로 수정할 필요가 있음.
2. Info 및 Data의 값을 정해진 자료형으로 받아오고있기 때문에 해당하는 정보도 재사용성을 위한 템플릿화.