= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

전기프2 15조 기획서 초고

201812767 이지훈

= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

1. **객체 생성 및 예외처리 :: 클래스 구조**

* // cDate : 조부모 클래스
* class cDate
* 일정의 타입, 날짜, 시간, 반복여부, 반복주기, 윤년 여부 등 date에 관련된 멤버 변수와 함수를 가지고 있습니다. Schedule의 조부모 클래스로서 해당 데이터를 Schedule 클래스에 넘겨줍니다.
* // cUser : 부모 클래스
* class cUser :

public cDate

* KU\_do\_list의 사용자인 User의 ID와 PW를 멤버 변수로 가지고 있으며, 사용자의 ID를 볼 수 있습니다. Schedule 클래스의 부모 클래스입니다. 마찬가지로 해당 데이터를 Schedule 클래스에 넘겨줍니다.
* // cSchedule : 자식 클래스
* class cSchedule :

public cUser

* 카테고리를 저장 및 분류, 스케줄 이름, 스케줄 완료 여부 데이터 및 위에서 상속받은 데이터들을 모두 가지고 있는 실질적 사용 객체입니다. 1개의 Schedule 객체는 1개의 일정에 관한 모든 데이터 및 함수들을 가지고 있습니다.

= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

1. **객체생성 및 예외처리 :: 데이터 구조**

* cDate : 조부모 클래스

private:

// @ 12달의 각 일수 저장 (윤년 계산)

int\* daysofMonth;

// @ 반복 일정의 유형 별 data

int repeatAnnual = -1;

int repeatMonthly = -1;

int repeatWeekly = -1;

protected:

// @ 일정의 타입

int sType = -1;

/\*

- sType = 0 :: 단발성 일정 (반복x)

- sType = 1 :: 기간한정 일정 (반복x)

- sType = 2 :: 반복성 일정 (매 년 같은 월)

- sType = 3 :: 반복성 일정 (매 월 같은 날)

- sType = 4 :: 반복성 일정 (매 주 같은 요일)

\*/

// @ 일정의 시작 시각

int sYear, sMonth, sDay, sHour, sMin, sSec;

// @ 일정의 종료 시각 ( case of isSingle = false )

int eYear, eMonth, eDay, eHour, eMin, eSec;

// @ 윤년 체크

bool isLeafYear = false;

* cUser : 부모 클래스

private:

// @ User 이름

string uName;

// @ User 권한 On/Off

bool isUser = false;

// @ User ID 및 PW (추후 구현할 로그인 기능)

string uId, uPw;

* protected가 아닌 private 멤버 변수로 설정한 것은 User의 ID와 PW를 쉽게 변경할 수 없도록 하기 위함으로써, getUserName() 등의 다른 protected type의 메소드를 이용해 접근할 수 있도록 하였습니다.
* cSchedule : 자식 클래스

private:

// @ 카테고리 데이터, 카테고리 수

string\* categoryData = nullptr;

int categoryNum;

// @ 스케줄 이름

string sName;

// @ 스케줄 완료 여부

bool sIsDone = false;

= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

1. **객체생성 및 예외처리 :: 메소드**

* 객체 생성 예외처리
* 객체 생성 시 먼저 일정의 type을 판단하고, type 선택 시 올바른 선택을 하지 않으면 처음으로 돌아가 다시 입력하게 합니다. 입력값에 따라 판단한 타입이 0~4 사이의 값으로 올바르다면, 일정의 시작 (및 종료) 날짜를 공백을 기준으로 “YYYY MM DD HH MM SS” 형식으로 입력받습니다. (ex. 2022 10 1 15 30 0). 해당 입력 양식이 지켜지지 않은 입력이 주어지거나, 날짜 및 시각의 범위에 벗어난 경우 (타입 체크 후인) 날짜 입력 단계로 돌아가서 다시 날짜를 입력하도록 하였습니다.
* main함수에서 Schedule \*s = new Schedule[100]; 을 하여 빈 일정 100개를 미리 만든 뒤, 새로운 일정을 추가할 때 0으로 초기화된 index의 값을 1씩 증가시키며 일정 1개 씩을 생성하도록 하였습니다. 100개를 초과하는 일정을 만들 수 없도록 하였습니다.