데이터설계

**[Car 클래스]**

class Car

{

private:

int userNum; //고객번호

int carNum; //차종코드

int defPay; //기본요금

int Km; //주행거리

int totPay; //총임대료

public:

//고객번호, 차종코드, 운행거리

void input(int un, int cn, int km);

//기본요금, 총요금 계산

void cal();

//대입연산자 멤버함수

Car& operator=(const Car &cr);

//set get시리즈

void setuserNum(int un);

int getuserNum();

void setcarNum(int cn);

int getcarNum();

void setdefPay(int dp);

int getdefPay();

void setKm(int km);

int getkm();

void settotPay(int tp);

int gettotPay();

};

**[Node 구조체]**

typedef struct \_node Node;

typedef struct \_node{

Node \*prev; //prev

Car cr; //Car클래스

Node \*next; //next

}Node;

**[companyA 클래스]**

class companyA

{

private:

Node \*headp; //headp

Node \*tailp; //tailp

Node \*curp; //curp

int len; //개수count

public:

companyA(); //생성자(create)

void display(); //출력

int searchUnique(int &unum); //사용자코드검색

Node\* appendFromTail(Car &kr); //테일노드앞에 추가

void sortList(); //사용자코드 오름차순정렬

int getlen(); //getlen

~companyA(); //소멸자(destroy)

};

함수설계

**[Car 멤버함수]**

|  |  |  |
| --- | --- | --- |
| **리턴값** | **함수명** | **기능** |
| void | input(int un, int cn, int km) | 고객번호, 차종코드, 운행거리 입력 |
| void | cal() | 차종별로 기본요금, 총요금계산  총요금(기본요금+주행거리\*운행거리당 임대료) |
| Car& | operator=(const Car &cr) | 대입연산자 멤버함수 |
| void | setuserNum(int un) | 고객번호 입력 |
| int | getuserNum() | 고객번호 리턴 |
| void | setcarNum(int cn) | 차종코드 입력 |
| int | getcarNum() | 차종코드 리턴 |
| void | setdefPay(int dp) | 기본요금 입력 |
| int | getdefPay() | 기본요금 리턴 |
| void | setKm(int km) | 운행거리 입력 |
| int | getkm() | 운행거리 리턴 |
| void | settotPay(int tp) | 총요금 입력 |
| int | gettotPay() | 총요금 리턴 |

**[companyA 멤버함수]**

|  |  |  |
| --- | --- | --- |
| **리턴값** | **함수명** | **기능** |
|  | companyA() | 헤드,테일노드 생성 len=0 |
| void | display() | 출력 |
| int | searchUnique(int &unum) | 사용자코드 검색(존재하면 return 1  아니면 return 0) |
| Node\* | appendFromTail(Car &kr) | 테일노드앞에 노드생성 |
| void | sortList() | 사용자코드로 오름차순 정렬 |
| int | getlen() | Len 리턴 |
|  | ~companyA() | 노드삭제 |