1. ReadKey와 클래스를 이용하여 달리는 자동차를 클래스로 만들어보자. 클래스에 자동차 정보, 메소드를 저장하고   
ReadKey를 이용해 자동차의 변화를 입력받는다.

1-1 차량번호, 속도, 방향을 필드로 가지는 car class를 선언하고 차량번호, 속도, 방향을 초기화 해주는 디폴트 생성자를 선언하라.

1-2 속도를 높이고 낮추는 speed\_up, speed\_down 메소드, 방향을 바꾸는 turn\_left, turn\_right 메소드를 선언하여라.  
키보드의 위 화살표를 누르면 속도가 늘어나고 아래 화살표는 속도가 줄어든다.   
왼쪽 화살표를 누르면 방향이 왼쪽으로 바뀌고 오른쪽 화살표를 누르면 방향이 오른쪽으로 바뀐다.

1-3 main 함수에서 클래스 객체를 만들어 임의의 차량번호, 속도, 방향을 가지도록 생성자를 호출한다.   
무한 루프에서 키를 계속 입력 받게 만들고 키가 눌렸을 때 해당 키에 해당하는 함수를 호출하고 결과를 보여라.

|  |
| --- |
| 출력 예시) |

class car

{

int carnum; //차번호

int speed; //속도

int dir; //방향

public car() { }

public car(int innum, int inspeed, int indir) { }

public void speedup() { }

public void speeddown() { }

public void turnright() { }

public void turnleft() { }

}

class Program

{

static void Main(string[] args)

{

car newcar = new car(1234, 70, 180);

ConsoleKeyInfo key;

//key.Key == ConsoleKey.UpArrow -> 위쪽 방향키가 눌렸다는 것을 알 수 있음.

}

}

2. 가계부를 만들어 보자. 클래스 이름은 account로 하여라.

2-1 클래스 account의 필드는 “순번, 날짜, 구분, 내용, 가격” 으로 구성되어 있다. main에서 데이터를 10개까지 저장   
할 수 있도록 클래스 배열로 만들어 객체를 생성하여라. 클래스 내에 날짜, 구분, 내용, 가격을 초기화 하는   
디폴트 생성자를 작성하여라. 또한, “구분” 과 “가격” 필드는 get, set 함수를 만들어라.

2-2 잔액은 main 함수에서 사용자로부터 입력받도록 작성하여라.

2-2 가계부에 값을 입력 하는 input\_account( ) 메소드를 작성하여라. Input\_account( ) 메소드 호출 시 “순번”은 사용자로부터 입력 받는 것이 아니며, count 변수를 사용하여 자동으로 입력 되도록 작성하여라.

2-3 가계부에 입력된 “가격” 정보를 수정하는 mod\_account( ) 메소드를 작성하여라. 날짜, 구분, 내용은 수정하지 않는다.

2-4 가계부에 입력되어 있는 값을 출력하는 print\_account( ) 메소드를 작성하여라.   
입력된 전체 값을 출력 한 후에 “총 수입”, “총 지출”, “남은 돈” 을 함께 출력하여라.

|  |
| --- |
| 출력 예시) |

class account

{

private int num; //순번

private string date; //날짜

private string separation; //구분

private string contents; //내용

private int money; //가격

}

public string Separation

{

get { return separation; }

set { separation = value; }

}

public int Money

{

get { return money; }

set { money = value; }

}

static void Main(string[] args)

{

int total\_money; //잔액

int import = 0; //총 수입 금액

int spending = 0; //총 지출 금액

//객체를 생성 하고, 인스턴스화 시켜줌

account[] my\_account = new account[10];

for (int i = 0; i < 10; i++)

{

my\_account[i] = new account();

}

Console.Write("잔액 : ");

total\_money = Convert.ToInt32(Console.ReadLine());

do

{

Console.Write("선택하세요 : (1 : 입력, 2 : 수정, 3 : 출력, 4 : 종료) >> ");

result = Convert.ToInt32(Console.ReadLine());

switch (result)

{

case 1:

break;

case 2:

break;

case 3:

break;

default:

break;

}

} while (result != 4);

}

3 . 클래스를 이용해서 냉장고를 만들어보자. 모든 함수 및 변수는 클래스 내에 구현하며 사용자로부터 어떤 작업을 할 지 입력받고 이를 처리하는 부분만 메인 함수에 작성하라. (1번 항목 입력, 2번 항목 삭제, 3번 출력, 4번 온도 변경, 5번 종료)

3-1 현재 온도를 저장하는 정수형 변수 curr\_temp, 물품 리스트를 저장하는 1차원 배열 itemlist를 선언하고   
현재 온도와 itemlist를 초기화 하는 디폴트생성자를 정의하여라. 최대 물품 개수는 20개이다.

3-2 객체 생성시 현재 온도를 설정해 주는 생성자를 작성하여라.

3-3 물품 리스트를 추가하는 additem 함수를 작성하여라.   
또한 리스트에서 해당 물품이 있는지 확인하고, 존재한다면 삭제하는 delitem 함수를 정의하라. 물품 항목은 사용자로부터 입력 받고, 삭제 후 해당 항목의 뒤에 다른 항목이 존재한다면 이를 앞으로 당겨온다.

3-4 현재 온도와 존재하는 아이템의 개수를 출력하는 print 함수, 온도를 바꾸는 changetemp 함수를 정의하라.

|  |
| --- |
|  |

class refrigerator

{

int curr\_temp;

string[] itemlist = new string[20];

public refrigerator() { }

public refrigerator(int intemp) { }

public void additem() { }

public void delitem() { }

public void print() { }

public void changetemp() { }

}

static void Main(string[] args)

{

refrigerator myref = new refrigerator(-7); //객체 생성시 냉장고 온도 설정

int input;

do

{

Console.Write("1 : 항목 입력, 2 : 항목 삭제, 3 : 출력, 4 : 온도 변경, 5 : 종료 : ");

input = Convert.ToInt32(Console.ReadLine());

switch (input)

{

case 1:

myref.additem();

break;

case 2:

myref.delitem();

break;

case 3:

myref.print();

break;

case 4:

myref.changetemp();

break;

case 5:

Console.WriteLine("exit");

break;

default:

//donothing

break;

}

} while( input != 5 );

}