객체지향 응용 프로그래밍

12201928 이상혁

|  |  |  |  |
| --- | --- | --- | --- |
| 9주차 HW – 연산자 오버로딩 (1) | | | |
| ID | 12201928 | Name | 이상혁 |
| Problem | | | |
| **Chapter 11:**  **연산자 오버로딩 (1)**  **1. OBJECTIVE (실습 목적)**   * 연산자 오버로딩 사용법을 익힌다.   **2. SCOPE (실습 범위)**   * 연산자 오버로딩 기본사항 * 연산자 오버로딩 제한조건 * 멤버/전역 함수로의 연산자 함수 * 연산자 오버로딩과 stream 연산자 * 단항 연산자 * 이항 연산자   **3. EXERCISES (실습 예제)**   * Lecture note에 수록되어 있는 예제를 실습하시오.   **4. PRACTICES (실습 과제)**  *실습과제에는 반드시 이론수업에서 배운 내용이나 개념이 적용되어야 함*  (Problem 1)   * 주어진 “Rational Number 헤더파일”을 이용하여 다음 기능을 제공하는 클래스 RationalNumber(분수)를 작성하라.   + 분수에서 분모가 0인 경우를 방지하고 약분하지 않은 분수를 약분하거나 간단히 만들고 음수 분모를 피하는 생성자를 작성하라.   + 이 클래스의 덧셈, 뺄셈 곱셈, 나눗셈 연산자를 오버로딩하라.   + 이 클래스의 관계 및 동일 연산자를 오버로딩 하라.   Rational Number 헤더파일:   |  | | --- | | #ifndef RATIONAL\_NUMBER\_H  #define RATIONAL\_NUMBER\_H  class RationalNumber {  public:  RationalNumber( int = 0, int = 1 ); // default constructor  RationalNumber operator+( const RationalNumber& ); // addition  RationalNumber operator-( const RationalNumber& ); // subtraction  RationalNumber operator\*( const RationalNumber& ); // multiplication  RationalNumber operator/( RationalNumber& ); // division    // relational operators  bool operator>( const RationalNumber& ) const;  bool operator<( const RationalNumber& ) const;  bool operator>=( const RationalNumber& ) const;  bool operator<=( const RationalNumber& ) const;  // equality operators  bool operator==( const RationalNumber& ) const;  bool operator!=( const RationalNumber& ) const;  void printRational() const; // display rational number  private:  int numerator; // private variable numerator  int denominator; // private variable denominator  void reduction(); // function for fraction reduction  }; // end class RationalNumber  #endif | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 9주차 HW – 연산자 오버로딩 (1) | | | |
| ID | 12201928 | Name | 이상혁 |
| Solutions(Source code and results) | | | |
| **헤더파일**    **Cpp 구현 파일**        **드라이버 파일**      코드 설명은 주성으로 해두었다.  **Result**    결과설명:  처음에는 드라이버 파일에 입력한 두 분수를 출력하였다.  이후 연산자 오버리딩을 통한 연산들이 정상적으로 이뤄지는 모습을 확인할 수 있다.  대소비교가 정상적으로 수행되는 모습을 확인할 수 있다. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 9주차 HW – 연산자 오버로딩 (1) | | | |
| ID | 12201900 | Name | 백재윤 |
| Discussions | | | |
| **이번주차는 연산자 오버로딩을 통해 사칙연산과 대소비교를 하는 과제였다.**  **예전 ratinal 과제랑 비슷해 생각보다 수월하게 할 수 있었다. 오버로딩의 개념을 이해하고 다시 복습하며 정상적으로 출력되게 하였다.**  **이를 이용한다면 드라이버 파일에서 사용자에게 직접 분수를 입력받고 원하는 기능을 선택하게 하여 좀더 구체적인 프로그램을 제작할 수 있을거같다는 생각이 들었다.**  **객체와 클래스에 한걸음 다가간 기분이였으며 stream연산자도 공부할 수 있었던 기회였다.**  **앞으로의 과제에서는 조금 더 열심히 임해야겠다.** | | | |