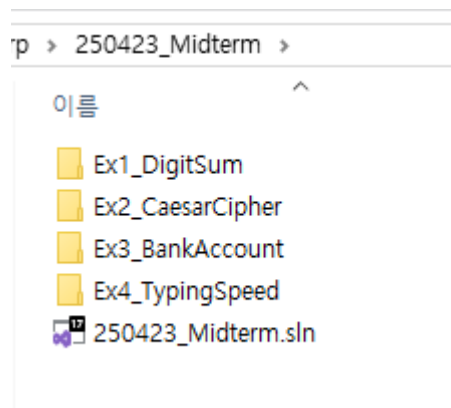


## 객체지향 프로그래밍 중간고사

1. 빈 솔루션 생성하기 -> 250423\_Midterm
2. 각 문제마다 프로젝트 생성하기. 문제마다 프로젝트 명이 작성되어 있음.  
해당 프로젝트 명으로 생성 할 것
3. 문제 풀이 완료 후 압축 시 .vs, bin폴더 삭제 후 압축하여 제출할 것.
4. 제출 구조는 솔루션 파일과 프로젝트 폴더들이 함께 나란히 있는 구조.  
폴더 안쪽에 개별 프로젝트 소스코드가 들어가야함.
5. 위 사항을 지키지 않을 시 감점 함.



## 문제1) 각 자리수 더하기 : Ex1\_DigitSum (5점)

사용자로부터 양의 정수를 입력받아, 해당 정수의 각 자리수를 모두 더한 결과를 출력하는 프로그램을 작성하세요.

### 요구사항:

1. 사용자로부터 양의 정수를 입력받습니다.
2. 입력받은 정수의 각 자리수를 모두 더합니다.
3. 합계를 출력합니다.
4. 입력이 양의 정수가 아닌 경우, 적절한 오류 메시지를 출력합니다.

### 입출력 예시:

정수를 입력하세요: 123  
각 자리수의 합:  $1 + 2 + 3 = 6$

정수를 입력하세요: 9528  
각 자리수의 합:  $9 + 5 + 2 + 8 = 24$

정수를 입력하세요: -123  
양의 정수를 입력해주세요.

## 문제2) 간단한 문자열 암호화 Ex2\_CaesarCipher (5점)

사용자로부터 문자열과 이동할 값(시프트 값)을 입력받아, 시저 암호(Caesar cipher) 방식으로 암호화하는 프로그램을 작성하세요.

### 요구사항:

1. 사용자로부터 영문 문자열과 1-25 사이의 정수(시프트 값)를 입력받습니다.범위가 넘어갈 경우 오류메세지 출력
2. 각 알파벳을 시프트 값만큼 알파벳 순서상 뒤로 이동시켜 암호화합니다.
3. 알파벳만 암호화하고, 공백이나 특수문자는 그대로 유지합니다.
4. 대문자는 대문자로, 소문자는 소문자로 유지합니다.
5. 알파벳이 Z(z)를 넘어가면 다시 A(a)부터 시작합니다.

### 입출력 예시:

암호화할 문자열을 입력하세요: Hello World

시프트 값을 입력하세요(1-25): 3

암호화된 문자열: Khoor Zruog

### 문제3) 은행 계좌 관리 시스템 Ex3\_BankAccount (10점)

개인 계좌와 모임통장을 관리하는 은행 시스템을 구현하세요.

1. BankAccount 클래스를 생성하세요. 이 클래스는 다음 멤버를 포함해야 합니다:
  - 계좌 소유자 이름
  - 계좌 번호
  - 잔액
  - 생성자, 입금 메서드, 출금 메서드
2. 위 클래스 내에 모임통장을 위한 멤버를 구현하세요:
  - 모임통장 잔액
  - 모임통장 참여자 명단(List<string>)
  - 모임통장에 입금하는 메서드
  - 모임통장에서 출금하는 메서드
  - 모임통장 정보를 출력하는 메서드
  - 모임통장에 참여자를 추가하는 메서드
3. 클래스 구현 시 다음 조건을 만족해야 합니다:
  - 잔액이 부족한 경우 출금이 불가능하며 적절한 메시지를 출력해야 합니다.
  - 입금 및 출금 시 거래 내역이 출력되어야 합니다.
  - 모임통장은 하나만 존재해야 합니다.
4. 이 클래스를 활용하여 다음 작업을 수행하는 Program 클래스의 아래의 Main 메서드를 동작하게 하시오. (다음장 확인)

```
static void Main(string[] args)

{
    // 개인 계좌 생성

    BankAccount account1 = new BankAccount("김철수", "1001");

    BankAccount account2 = new BankAccount("이영희", "1002");

    // 개인 계좌 입출금

    account1.Deposit(50000);

    account1.Withdraw(20000);

    // 모임통장 참여자 추가

    Console.WriteLine("[모임통장 참여자 추가]");

    BankAccount.AddParticipant("김철수");

    BankAccount.AddParticipant("이영희");

    Console.WriteLine();

    // 모임통장 입출금

    BankAccount.DepositToGroup(100000);

    BankAccount.WithdrawFromGroup(30000);

    // 계좌 정보 출력

    Console.WriteLine("[계좌 정보 출력]");

    account1.PrintAccountInfo();

    BankAccount.PrintGroupAccountInfo();

    // 추가 테스트 - 잔액 부족 시 출금 시도

    Console.WriteLine("[추가 테스트 - 잔액 부족]");

    account2.Deposit(5000);

    account2.Withdraw(10000); // 잔액 부족 메시지가 출력되어야 함
}
```

## 5. <결과>

```
Microsoft Visual Studio 디버그 콘솔

[개인 계좌 생성]
이름: 김철수, 계좌번호: 1001, 잔액: 0원

[개인 계좌 생성]
이름: 이영희, 계좌번호: 1002, 잔액: 0원

[입금]
김철수님의 계좌에 50000원이 입금되었습니다.
현재 잔액: 50000원

[출금]
김철수님의 계좌에서 20000원이 출금되었습니다.
현재 잔액: 30000원

[모임통장 참여자 추가]
김철수님이 모임통장에 참여했습니다.
이영희님이 모임통장에 참여했습니다.

[모임통장 입금]
모임통장에 100000원이 입금되었습니다.
현재 모임통장 잔액: 100000원

[모임통장 출금]
모임통장에서 30000원이 출금되었습니다.
현재 모임통장 잔액: 70000원

[계좌 정보 출력]
--- 개인 계좌 정보 ---
이름: 김철수
계좌번호: 1001
잔액: 30000원

--- 모임통장 정보 ---
참여자 수: 2명
참여자 명단: 김철수, 이영희
잔액: 70000원

[추가 테스트 - 잔액 부족]
[입금]
이영희님의 계좌에 5000원이 입금되었습니다.
현재 잔액: 5000원

[출금 실패]
잔액이 부족합니다. 현재 잔액: 5000원
```

## 문제4) 타이핑 속도 측정기 Ex4\_TypingSpeed (10점)

버튼, 타이머, 레이블, 텍스트박스만을 사용하여 간단한 타이핑 속도 측정 프로그램을 만드세요.

### 요구사항:

1. 다음 컨트롤을 폼에 배치하세요 (모양 다음장 및 프로그램 압축파일 참고)
  - 제목용 레이블 (타이틀) -> 굵게, 18pt
  - 타이핑할 문장을 표시할 레이블
  - 사용자가 타이핑할 텍스트박스
  - 시작 버튼과 재시작 버튼
  - 경과 시간을 표시할 레이블
  - 결과(타이핑 속도)를 표시할 레이블
  - 타이머 컨트롤 (눈에 보이지 않음)
2. 기능 구현:
  - 시작 버튼을 누르면 타이머가 시작되고, 사용자는 표시된 문장을 텍스트박스에 타이핑합니다.
  - 타이머는 0.1초마다 실행되어 경과 시간을 업데이트합니다.
  - 사용자가 정확하게 문장을 타이핑하면, 타이머가 멈추고 타이핑 속도(분당 타이핑 수)를 계산하여 표시합니다.
  - 재시작 버튼을 누르면 프로그램이 초기화됩니다.
  - 시작을 누르고 난 후
    - i. 시작버튼 Enabled = false
    - ii. 텍스트박스 Enabled = true
  - 시작 버튼 누르기 전
    - i. 시작버튼 Enabled = true
    - ii. 텍스트박스 Enabled = false
3. 설계 요소:
  - 타이핑할 문장은 간단한 것으로 미리 프로그램에 설정합니다.
  - 경과 시간은 초 단위로 표시합니다.
  - 타이핑 속도는 분당 문자 수(CPM)로 계산합니다.
4. 힌트
  - 텍스트박스를 지우는 방법 : 텍스트박스 객체의 Clear() 메서드
  - 시작 버튼 클릭 후 텍스트박스에 커서가 생기게 하는 방법 : 텍스트박스 객체의 Focus(); 메서드

Form1

—

□

×

## 타이핑 속도 측정기

C# 프로그래밍은 재미있고 유용한 기술입니다. 열심히 공부하면 다양한 프로그램을 만들 수 있습니다.

시작

재시작

경과시간 : 0.0초      타이핑 속도 : 0 CPM

Form1

—

□

×

## 타이핑 속도 측정기

C# 프로그래밍은 재미있고 유용한 기술입니다. 열심히 공부하면 다양한 프로그램을 만들 수 있습니다.

C# 프로그래밍은 재미있고 유용한 기술입니다. 열심히 공부하면 다양한 프로그램을 만들 수 있습니다.

시작

재시작

경과시간 : 13.4초

타이핑 완료

i

축하합니다! 타이핑 속도는 246 CPM입니다.

확인