

## **Esercizio: gestione di un conto corrente**

Si realizzi un conto corrente bancario mediante:

- una ASTRAZIONE DI DATO
- un TIPO DI DATO ASTRATTO

Il conto corrente è caratterizzato dalle seguenti informazioni:

- numero del cc;
- saldo;
- ultimi movimenti.

**FUNZIONI DA IMPLEMENTARE:**

- apertura: definizione del numero di conto e inizializzazione del saldo
- versamento: versamento di un certo importo sul conto
- prelievo: prelievo di un certo importo dal conto
- restituzione del saldo: restituisce il saldo attuale del conto
- stampa: stampa a video della lista delle operazioni effettuate sul conto (Movimento di 1000, Movimento di -800,...)

## Conto corrente: astrazione di dato

```
// File: cc.h

/*
Questo file header dichiara TUTTE e SOLE le funzioni
che possono essere usate per interagire con il conto
corrente. Solo le dichiarazioni presenti in questo
file header sono visibili "all'esterno" (ossia dagli
altri punti del programma) */

void apertura(int CCnum);
void versamento(int importo);
void prelievo(int importo);
int getSaldo();
void stampaMovim();
```

```

// File: cc.c

/* Questo file definisce un modulo "conto corrente",
in modo che sia possibile interagire con il conto
usando solo determinate funzioni e operazioni. */

#include <stdio.h>
#include "cc.h"

// Massimo numero di movimenti
// che si vuole memorizzare
#define MAX_MOV_SIZE 20

/*
Struttura che definisce il conto corrente. Siccome
viene dichiarata "static", puo' essere vista solo
all'interno di questo file => e' protetta a livello
di file. Grazie a questo livello di protezione e'
possibile definire funzioni/operazioni pubbliche che
agiscano sul dato protetto garantendo così la
coerenza dei dati */
static struct
{
    // numero di conto corrente
    int CCnum;
    // saldo
    int saldo;
    // ultimi movimenti
    int movimenti[MAX_MOV_SIZE];
    // quanti movimenti sono stati fatti sul conto
    int contMovim;
}cc;

```

```
////////////////////////////////////
```

```
/*  
  Restituisce il saldo del conto.  
*/  
int getSaldo()  
{  
  return(cc.saldo);  
}
```

```
/*  
  Apertura di un nuovo conto. Occorre specificare il  
  numero di conto da aprire  
*/  
void apertura(int CCnum)  
{  
  cc.CCnum = CCnum;  
  cc.saldo = 0;  
  cc.contMovim = 0;  
}
```

```
/*  
  Versamento sul conto. Specificare l'importo da  
  versare.  
*/  
void versamento(int importo)  
{  
  cc.saldo = cc.saldo+importo;  
  cc.movimenti[cc.contMovim]=importo;  
  cc.contMovim++;  
  printf("\nVersamento di Euro %d effettuato sul conto  
%d\n", importo, cc.CCnum);  
}
```

```

/*
  Prelievo: specificare l'ammontare.
*/
void prelievo(int importo)
{
  cc.saldo = cc.saldo-importo;
  cc.movimenti[cc.contMovim]=-importo;
  cc.contMovim++;
  printf("\nPrelievo di Euro %d effettuato sul conto
%d\n",importo,cc.CCnum);
}

/*
  Stampa gli ultimi movimenti.
*/
void stampaMovim()
{
  int i;

  printf("\nLista degli ultimi %d movimenti effettuati
sul conto %d:\n",MAX_MOV_SIZE,cc.CCnum);
  // ATTENZIONE: non devo comunque andare oltre
  // all'ultima locazione del vettore (MAX_MOV_SIZE)
  for(i=0;i<(cc.contMovim) && i<MAX_MOV_SIZE;i++)
    printf("Movimento di Euro
%d\n",cc.movimenti[i]);
}

```

```
// File: main1.c
/*
    Esempio di uso di cc.c
    Per compilare con gcc eseguire:
    gcc main1.c cc.c -o main1.exe
*/

#include <stdio.h>
#include "cc.h"

int main(void)
{

    apertura(3);
    versamento(10000);
    prelievo(1500);
    printf("\nIl saldo è:%d Euro\n",getSaldo());
    versamento(1200);
    stampaMovim();

    return 0;
}
```

## Conto corrente: tipo di dato astratto

```
// File: ccADT.h

/*
   Dichiarazione di conto corrente come tipo di dato
   astratto.
*/

// Massimo numero di movimenti
// che si vuole memorizzare
#define MAX_MOV_SIZE 20

/*
   Struttura che definisce il conto corrente. Siccome
   viene definito un tipo, e' possibile istanziare piu'
   variabili di tipo "conto corrente".
   Non vi e' nessuna protezione sui dati */
typedef struct
    {
        int CCnum;
        int saldo;
        int movimenti [MAX_MOV_SIZE];
        int contMovim;
    } CC;

/*
   Dichiarazioni che agiranno sul tipo di dato
   astratto. Notare come sia necessario usare un
   puntatore alla struttura CC.
*/
void apertura (CC *cc, int CCnum);
void versamento(CC *cc, int importo);
void prelievo(CC *cc, int importo);
int getSaldo(CC *cc);
void stampaMovim(CC *cc);
```

```

// File ccADT.c
/*
    Questo file definisce le operazioni che agiscono
    sul tipo di dato astratto. Si noti come in questo
    caso sia necessario passare un puntatore all'istanza
    di dato astratto.
*/

#include <stdio.h>
#include "CCadt.h"

int getSaldo(CC *cc)
{
    return cc->saldo;
}

void apertura (CC *cc, int CCnum)
{
    int i;

    cc->CCnum = CCnum;
    cc->saldo = 0;
    cc->contMovim = 0;
}

void versamento(CC *cc, int importo)
{
    cc->saldo = (cc->saldo)+importo;
    cc->movimenti[cc->contMovim] = importo;
    cc->contMovim++;
    printf("\nVersamento di Euro %d effettuato sul conto
%d\n", importo, cc->CCnum);
}

```



```

void prelievo(CC *cc, int importo)
{
    cc->saldo = (cc->saldo)-importo;
    cc->movimenti[cc->contMovim] = -importo;
    cc->contMovim++;
    printf("\nPrelievo di Euro %d effettuato sul conto
%d\n", importo, cc->CCnum);
}

```

```

void stampaMovim(CC *cc)
{
    int i;

    printf("\nLista degli ultimi %d movimenti effettuati
sul conto %d:\n", MAX_MOV_SIZE, cc->CCnum);
    for(i=0; i<(cc->contMovim) && i<MAX_MOV_SIZE; i++)
        printf("Movimento di Euro %d\n", cc-
>movimenti[i]);
}

```

```

// File: main2.c
/*
    Utilizzo del tipo di dato astratto conto corrente.
    Per compilare con gcc: gcc main2.c ccADT.c -o
main2
*/

#include <stdio.h>
#include "CCadt.h"

int main()
{
    // creazione di due dati astratti
    CC conto1,conto2;

    apertura(&conto1,1);
    apertura(&conto2,2);

    versamento(&conto1,1500);
    versamento(&conto2,1000);

    printf("\nSaldo del conto 1:
%d\n",getSaldo(&conto1));
    printf("\nSaldo del conto 2:
%d\n",getSaldo(&conto2));

    prelievo(&conto1,200);
    versamento(&conto2,500);

    printf("\nSaldo del conto 1:
%d\n",getSaldo(&conto1));
    printf("\nSaldo del conto 2:
%d\n",getSaldo(&conto2));

    stampaMovim(&conto1);
    stampaMovim(&conto2);
}

```