

Технически университет – София

Факултет по Компютърни системи и технологии

Курсова работа по дисциплина ПИК 1

Разработил: Красимир Руменов Марков Проверил: Гл. асистент Факултетен номер: 501220020 д-р Бюлбюл Зюлямова

Специалност: ИТИ, група 49

Подпис:....................

**Съдържание**

1. Задание за курсова работа
2. Описание на функционалността на кода
3. Блок схема
4. Source код на програмата

14. Курсова работа по дисциплина ПИК1 за студенти ФКСТ

Вие сте програмист в банка и трябва да направите софтуер за менажиране на сметки в банката. Вашата програма започва с въвеждане на съществуващите сметки(команда DATA и се въвежда до подаване на 3 числа – 0 0 0), като всяка смета има ID – цяло число и парола – 6 цифрен код. Всяка сметка има и сума по сметката – реално число.

След като въведете всички сметки имате няколко режима на работа.   
LOGIN – пита ви за номер на сметка и парола. Не може да продължите работата с програмата без да сте влезли в конкретна сметка.

LOGOUT – излиза от текущата сметка.

DRAW – пита ви за сума и я изтегля от сметката, ако сумата е налична. Ако не е налична да изведе подходящо съобщение.

PUT – внасяте пари в сметката ви.

TRANSFER – пита ви за сметка и ако тя съществува пренасяте сума от вашата сметка в чуждата.

INFO – показва баланса по вашата сметка.

EXIT – излиза от програмата

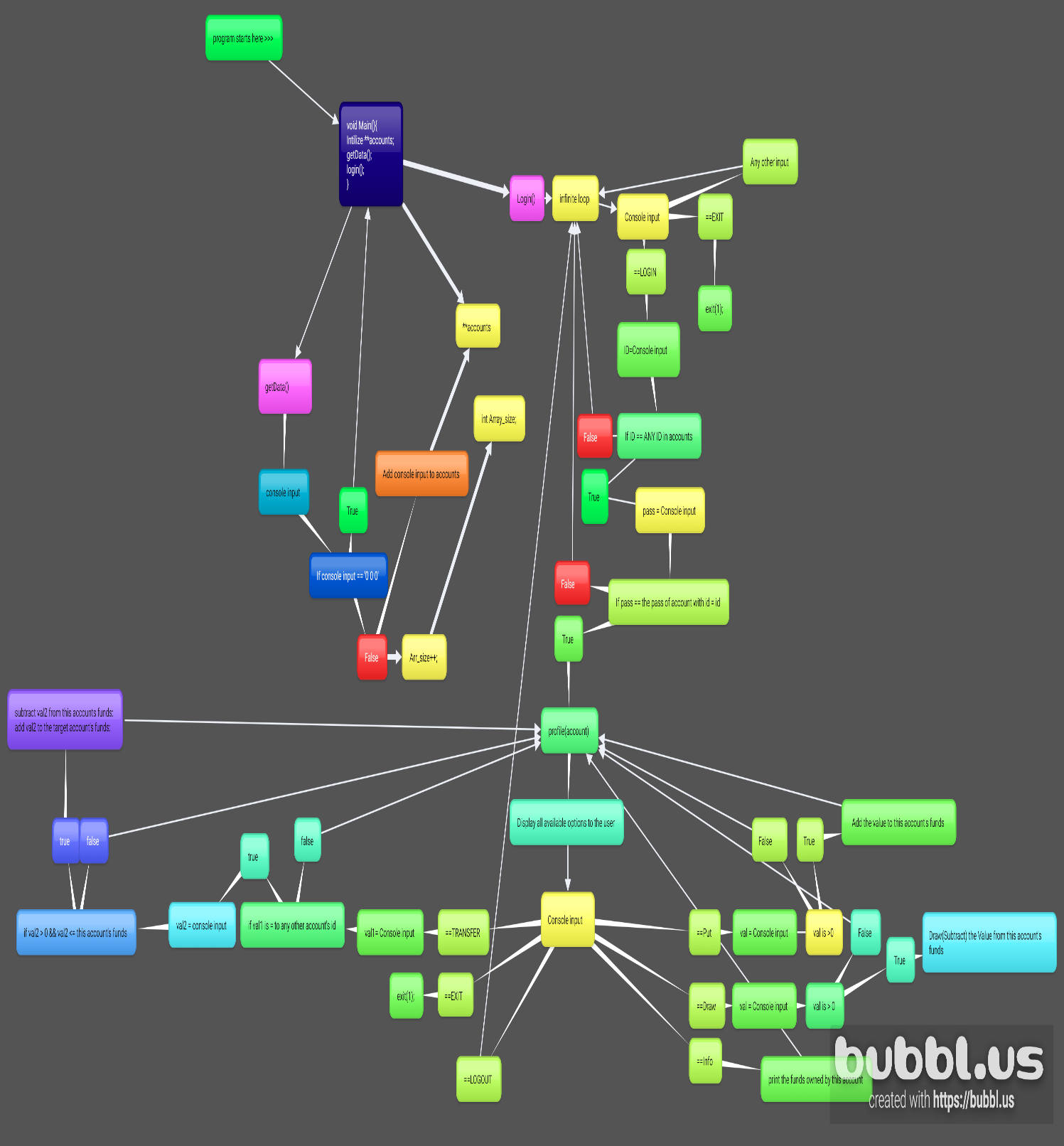
Студент:……………………                                  асистент:……………………

**Описание на функционалността на кода**

Програмата се състой от 7 функций.

1. „main” - състой се от for цикъл и деклариране на 2 функции (№2 и №3), също тaка проверка дали има достатъчно памет на устройството на, което се изпълнява.
2. “getData” - фунцията взима Input(ID , Password, Balance) от потребителя и и попълва двумерен масив до въвеждане на 0 0 0.
3. “login” - използват се два for цикъла за да се провери дали въведените данни съвпадат със някой от въведените в “getData” и влизаме в фунция №4
4. “profile” – изписва съобщение от възможни опции от менюто, като всяка опция се прочита като масив от “char” и се сръвнява със предварително заявани масиви от тип “char” от където постъпваме във фунции №5,№6 и №7, също така съдържа “info”, “logout” и “exit” опции, като “info” изписва настоящият баланс по сметката, „logout” връща на функция №3,а “exit” затваря програмата.
5. „put” – фунцията добавя сума към конкретната сметка достъпена с фунция №3.
6. “draw”- фунцията изважда сума от конкретната сметка достъпена с фунция №3.
7. “transfer”- фунцията изважда сума от конкретната сметка и чрез подаване на ID увеличава сумата в подаденото ID

**Блок схема**



**Source код на програмата**

**Github link:** **https://github.com/k-markov01/bank-UI**

*#include* <stdio.h>

*#include* <stdlib.h>

*#include* <string.h>

int arr\_size, i, j;

int\*\* accounts = NULL; *//int \*\* (Pointer to (array) of pointers to ints)*

void getData();

void login();

void profile();

void transfer();

void put();

void draw();

int main(void){

    printf("All console input should be CAPITAL letters or numbers\n");

*//The ammount of accounts cable of being stored is = to rows.*

*//Possible further optimization with realloc and dynamic creation of the accounts pointer array in the getData() function. ~ Not used in order to simplify the program.*

    int rows= 1000, colm=3;

*//Make a 2d Array of int pointers*

    accounts = malloc(rows \* sizeof(int\*));

*if* (accounts == NULL){

        fprintf(stderr, "Couldn't allocate memory[i][1].");

        exit(1);

    }

*for*(int i =0; i<rows ;i++){

        accounts[i] = malloc(colm \* sizeof(int));

*if* (accounts[i] == NULL){

            fprintf(stderr, "Couldn't allocate memory[i][2].");

            exit(1);

        }

    }

*//Add data to the 2d Array*

    getData(accounts, rows);

    printf("\nAccounts created: %d\n", arr\_size);

*//Log into an account*

    login(accounts);

*//In case something goes wrong to prevent memory leak.*

    free(accounts);

*return* 0;

}

void getData(int \*\*accounts, int rows){

    printf("All accounts accept ONLY integers for ID, password and funds.\nInput should look something like this: \">>>A B C\"\n");

    printf("Where \"A\" is the Id of the account, \"B\" is the password of the account, and \"C\" is the funds of the account.\n");

*//Insert data into the pointer arrays until a value of "0 0 0 is intered" or the array pointer size reaches the value of rows*

*for*(int i = 0; i < rows  ;i++){

        int id, pass, funds;

        printf("\nInstert ID, Password and Balance for account #%d.\n>>>", arr\_size+1);

        scanf(" %d%d%d", &id, &pass, &funds);

*if* (id == 0 && pass == 0 && funds == 0){

*return*;

        }

        accounts[i][0] = id ;

        accounts[i][1] = pass;

        accounts[i][2] = funds;

        arr\_size++;

    }

}

void login(int \*\*accounts){

    int i;

*for*(;;){

        int id, pass;

        printf("\nID to log into\n>>>");

        scanf("%d", &id);

*for* (i = 0; i < arr\_size; i++){

*if* (accounts[i][0] == id){

                printf("\nPASSWORD for account with id %d\n>>>", id);

                scanf("%d", &pass);

*if* (accounts[i][1] == pass){

                    profile(accounts[i]);

*break*;                }

*else* printf("Password is wrong"); *return*;

            }

*if* (i == arr\_size-1){

                printf("None of the id's match");

*return*;

            }

        }

        printf("\n\n");

    }

}

void profile(int \*account){

    printf("\nPossible optioins:\tPUT\t DRAW\t TRANSFER\t INFO\t LOGOUT\t EXIT\n>>>");

    char u\_in[20];

    char \_put[] = "PUT";

    char \_draw[] = "DRAW";

    char \_transfer[] = "TRANSFER";

    char \_info[] = "INFO";

    char \_logout[] = "LOGOUT";

    char \_exit[] = "EXIT";

    scanf("%s", &u\_in);

*if*(strcmp(u\_in, \_put) == 0) put(account);

*else* *if* (strcmp(u\_in, \_draw) == 0) draw(account);

*else* *if* (strcmp(u\_in, \_transfer) == 0) transfer(account);

*else* *if* (strcmp(u\_in, \_info) == 0){

        printf("Current balance is: %d\n", account[2]);

    }

*else* *if* (strcmp(u\_in, \_logout) == 0) *return*;

*else* *if* (strcmp(u\_in, \_exit) == 0) exit(0);

*else* printf("Input is incorect. Please try again.\n");

    profile(account);

}

void put(int \*account){

    int moneyInput;

    printf("Please enter the amount you wish to Input: \n>>>");

    scanf("%d", &moneyInput);

*if*(moneyInput <0){

        printf("The amount to Input cannot be less than 0. Please try again.\n");

*return*;

    }

    account[2] += moneyInput;

    printf("Current balance: %d\n", account[2]);

}

void draw(int \*account){

    int moneyDraw;

    printf("Please enter the amount you wish to Draw: \n>>>");

    scanf("%d", &moneyDraw);

*if* (moneyDraw > account[2]){

        printf("The amount cannot be greater than your current balance.\n");

*return*;

    }

*if*(moneyDraw <0){

        printf("The amount to draw cannot be less than 0. Please try again.\n");

*return*;

    }

    account[2] -= moneyDraw;

    printf("Current balance: %d\n", account[2]);

}

void transfer(int \*account){

    printf("Arr\_size is :%d\n",arr\_size);

    int i,j,targetID,moneyTransfer;

    printf("Possible ID's to Transfer:\n");

*for* (j=0;j<arr\_size;j++){

*if* (account == accounts[j]) *continue*;

        printf("%d\n",accounts[j][0]);

    }

    printf("Please enter the ID of the account you wish to Transfer to. \n>>>");

    scanf("%d", &targetID);

*for* (i=0;i<arr\_size;i++){

*if* (targetID == accounts[i][0]){

            printf("Enter the amount you wish to Transfer:");

            scanf("%d",&moneyTransfer);

*if* (moneyTransfer > account[2]){

                printf("Balance in your account isn't enough");

*return*;

            }

*else* *if* (moneyTransfer < 0){

                printf("The amount cannot be less then 0");

*return*;

            }

*else*{

                account[2]-=moneyTransfer;

                accounts[i][2] +=moneyTransfer;

                printf("You Trasfered %d to account ID:%d ",moneyTransfer,accounts[i][0]);

            }

        }

    }

}