TALLINNA TEHNIKAÜLIKOOL
INFOTEHNOLOOGIA TEADUSKOND
Arvutisüsteemide instituut

 xxxxxxxxxxxxxxx yy3083llll

**FAILID + KIRJED**

Kodutöö nr. 1

Juhendaja: dotsent Vladimir Viies

Tallinn 20xx

# AUTORIDEKLARATSIOON

Kinnitan, et käesolev töö on minu töö tulemus ja seda ei ole minu ega kellegi teise poolt varem esitatud.

xxxxxxxxxxxxxxxxxxx

AUTHOR'S DECLARATION

I certify that this work is the result of my own work and has not been previously submitted by me or anyone else.

SISUKORD

[AUTORIDEKLARATSIOON 2](#_Toc351927687)

[ÜLESANDE PÜSTITUS 4](#_Toc351927688)

[PILT ALGORITMIST 5](#_Toc351927689)

[FAILIDE SISU 6](#_Toc351927689)

[PROGRAMMIKOOD](#_Toc351927689) 7, 8

[PROGRAMMI SELETUS](#_Toc351927689) 9

[PILT PROGRAMMIST](#_Toc351927689) 10

[KASUTATUD KIRJANDUS](#_Toc351927717) 11

**TABLE OF CONTENTS**

**AUTHOR'S DECLARATION....................................................................... 2**

**SETTING UP THE TASK .............................................................................4**

**PICTURE OF THE ALGORITHM .............................................................5**

**CONTENT OF FILES................................................................................... 6**

**PROGRAM CODE.................................................................................... 7, 8**

**PROGRAM EXPLANATION..................................................................... 9**

**PICTURE OF THE PROGRAM................................................................ 10**

**REFERENCES............................................................................................. 11**

# ÜLESANDE PÜSTITUS

Variant: K-9

Koostada algoritm ja sellele üksüheselt vastav programm (C -keeles), mille abil:

1. failist F1(Input\_name\_code.txt) sisestatakse kirjed

Tudeng struktuuriga:

• Nimi – string,

• Kood (üliõpilase kood) kümnetäheline;

2. failist F2(Input\_exam\_results.txt) sisestatakse eksami tulemused:

• kirjed Aine struktuuriga:

 -Ainenimetus – string,

 -Tud (tudengikood) – string,

 -Hinne – naturaalarvuline (0..5)

3. faili F3(Output.txt) väljastatakse täidetud eksamileht.

 SETTING UP THE TASK

Variant: K-9

Create an algorithm and a corresponding program (in C language) that will:

1. Records are entered from file F1(Input\_name\_code.txt).

A student with structure:

• Name – string,

• Code (student code) ten letters;

2. from the file F2(Input\_exam\_results.txt) the exam results are entered:

• records with Substance structure:

 -Subject name – string,

 -Tud (student code) – string,

 -Grade - natural number (0..5)

3. The completed exam sheet is issued to file F3(Output.txt).

# PILT ALGORITMIST

# FAILIDE SISU

**Input\_name\_code.txt**

Albert Laiuste 143083IASB

Martti Orr 120878AAVB

Sven Koiv 142799IAPB

Silver Kurrik 155556IAPB

Ott Puusepp 148529IABB

Aleks Savioja 156742AASB

**Input\_exam\_results.txt**

Programmeerimine 143083IASB 4

Fuusika 120878AAVB 5

Fuusika 142799IAPB 1

Fuusika 143083IASB 5

Programmeerimine 142799IAPB 5

Side 120878AAVB 4

Side 143083IASB 3

Side 142799IAPB 2

Side 155556IAPB 0

Fuusika 148529IABB 3

Fuusika 156742AASB 4

**Output.txt**

Nimi: Albert Laiuste

Aine nimetus: Tudengikood: Hinne:

Programmeerimine 143083IASB 4

Fuusika 143083IASB 5

Side 143083IASB 3

Nimi: Martti Orr

Aine nimetus: Tudengikood: Hinne:

Fuusika 120878AAVB 5

Side 120878AAVB 4

Nimi: Sven Koiv

Aine nimetus: Tudengikood: Hinne:

Fuusika 142799IAPB 1

Programmeerimine 142799IAPB 5

Side 142799IAPB 2

Nimi: Silver Kurrik

Aine nimetus: Tudengikood: Hinne:

Side 155556IAPB 0

Nimi: Ott Puusepp

Aine nimetus: Tudengikood: Hinne:

Fuusika 148529IABB 3

Nimi: Aleks Savioja

Aine nimetus: Tudengikood: Hinne:

Fuusika 156742AASB 4

# PROGRAMMIKOOD

/\*\*

 \* Fail: hw1\_143083IASB.c

 \* Autor: Mart xxxxx

 \* Created: 10.02.20xx

 \* Modified: 11.02.20xx

 \*

 \* Variant: K-9

 \*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define INPUT\_NAME\_CODE "Input\_name\_code.txt"

#define INPUT\_CODE\_RESULTS "Input\_exam\_results.txt"

#define OUTPUT\_FILE "Output.txt"

**typedef** struct **{**

 char firstName**[**32**];**

 char lastName**[**32**];**

 char studentCode1**[**256**];**

 char studentCode2**[**256**];**

 char lessonName**[**32**];**

 int grade**;**

**}** Details**;**

int inputControl**()**

**{**

 FILE **\***inputNameCode **=** fopen**(**INPUT\_NAME\_CODE**,** "r"**);**

 FILE **\***inputCodeResults **=** fopen**(**INPUT\_CODE\_RESULTS**,** "r"**);**

 **if** **((**inputNameCode **==** **NULL)** **||** **(**inputCodeResults **==** **NULL))** **{**

 printf**(**"Input failid puudu.\n"**);**

 **return** **-**1**;**

 **}**

 **else{**

 fclose**(**inputNameCode**);**

 fclose**(**inputCodeResults**);**

 **return** 0**;}**

**}**

void process**()**

**{**

 Details person**[**256**];**

 int i**=**0**,**n**=**0**,**j**=**0**,**k**=**0**;**

 //Avab töötluseks failid

 FILE **\***outputResults **=** fopen**(**OUTPUT\_FILE**,** "w"**);**

 FILE **\***inputNameCode **=** fopen**(**INPUT\_NAME\_CODE**,** "r"**);**

 FILE **\***inputCodeResults **=** fopen**(**INPUT\_CODE\_RESULTS**,** "r"**);**

 //Loeb sisse Eesnimi Perenimi Tudengikood

 **while(**fscanf**(**inputNameCode**,** "%s %s %s\n"**,** person**[**i**].**firstName**,** person**[**i**].**lastName**,** person**[**i**].**studentCode1**)** **!=** EOF**){**

 i**++;**

 n**++;**

 **}**

 //Loeb sisse Ainenimi Tudengikood Hinne

 **while(**fscanf**(**inputCodeResults**,** "%s %s %d\n"**,** person**[**j**].**lessonName**,** person**[**j**].**studentCode2**,** **&**person**[**j**].**grade**)** **!=** EOF**){**

 j**++;**

 k**++;**

 **}**

 **for(**i**=**0**;** i**<**n**;** i**++){**

 **if** **(**i**!=**0**)** **{**fprintf**(**outputResults**,** "\n"**);}**

 fprintf**(**outputResults**,** "Nimi: %s %s\n"**,** person**[**i**].**firstName**,** person**[**i**].**lastName**);**

 fprintf**(**outputResults**,** "Aine nimetus: Tudengikood: Hinne:\n"**);**

 **for** **(**j**=**0**;** j**<**k**;** j**++){**

 **if(**strcmp**(**person**[**i**].**studentCode1**,** person**[**j**].**studentCode2**)** **==** 0**)** **{** //Vaatab, kas tudengikoodid on samasugused

 fprintf**(**outputResults**,** "%-16s %-15s %3d\n"**,** person**[**j**].**lessonName**,** person**[**j**].**studentCode2**,** person**[**j**].**grade**);**

 **}**

 **}**

 **}**

 //Sulgeb avatud failid

 fclose**(**inputNameCode**);**

 fclose**(**inputCodeResults**);**

 fclose**(**outputResults**);**

**}**

int main**(**void**)**

**{**

 //Kontrollib kas vajalikud failid eksisteerivad

 **if** **(**inputControl**()** **!=** 0**){**

 exit**(**1**);}**

 process**();**

 **return** 0**;**

**}**

# PROGRAMMI SELETUS

Programm koosneb peaprogrammist int main**()**, kahest alamprogrammist
void inputControl**()**,void process**()**.

Alamprogrammis void inputControl**()** proovitakse avada kontrolli eesmärgil määratud input failid. Kui kontroll õnnestub tagastatakse vastus 0, mis tähendab, et kõik failid avanesid korralikult. Kui programm ebaõnnestub siis tagastatakse vastus -1.

Alamprogrammis void process**()** avatakse esmalt töötluseks kõik 3 tekstifaili, seejärel loetakse input failide väärtused mällu struktuuridesse. Kuna mõlemas failis on ühine väärtus tudengikood(studentCode), siis neid võrreldakse ja hiljem väljastatakse väärtused mis on omavahel seotud ülesande püstituses kirjeldatud nõudmiste järgi Output.txt faili.

Peaprogramm int main**()** jooksutatakse inputControl**()** kontrolli esimese asjana ja kui tagastatud väärtus erineb 0-st siis edasist tööd ei teha ja programm suletakse. Kui failid eksisteerivad jooksutatakse järgminena void process**()**ja kui see töötas veatult siis programm suletakse.

EXPLANATION OF THE PROGRAM

**The program consists of the main program int main(), two subprograms**

**void inputControl(), void process().**

**The void inputControl() subroutine tries to open the specified input files for control purposes. If the check succeeds, the answer 0 is returned, which means that all files were opened properly. If the program fails, the answer -1 is returned.**

**In the void process() subroutine, all 3 text files are first opened for processing, then the values of the input files are read into memory structures. Since both files have a common value studentCode (studentCode), they are compared and later the values that are related to each other according to the requirements described in the assignment statement are output to the Output.txt file.**

**The main program int main() is run as the first thing in the inputControl() check, and if the returned value is different from 0, no further work is done and the program is closed. If the files exist, void process() is run next, and if it worked flawlessly, the program is closed.**

# PILT PROGRAMMIST PICTURE OF THE PROGRAM

# (Silumisrežiimis)( in debug mode)



# KASUTATUD KIRJANDUS

<http://www.tud.ttu.ee/im/Vladimir.Viies/materials/> .......