Картина, която съдържа текст, графична колекция

Описанието е генерирано автоматичноТЕХНИЧЕСКИ УНИВЕРСИТЕТ – ВАРНА

Факултет по изчислителна техника и автоматизация

Катедра „Софтуерни и интернет технологии“

**СЕМЕСТРИАЛНА ДОМАШНА РАБОТА**

по дисциплината „Базово програмиране”

на тема: „Информационна система Студенти”

Вариант 241

|  |  |
| --- | --- |
| Изготвил: Гилгамеш Алджабари | Проверил: |
| Специалност: СИТ |  |
| Група: 3 |  |
| Факултетен номер: 23621654 |  |

2024

Съдържание

[I. Задание на проекта 3](#_Toc120821980)

[II. Анализ на решението 4](#_Toc120821981)

[1. Структура за данните в програмата 4](#_Toc120821982)

[2. Реализация на условие A 4](#_Toc120821983)

[3. Реализация на условие B 4](#_Toc120821984)

[4. Реализация на условие C 5](#_Toc120821985)

[5. Реализация на условие D 5](#_Toc120821986)

[6. Реализация на условие E 6](#_Toc120821987)

[7. Реализация на условие F 7](#_Toc120821988)

[8. Реализация на условие … - допълнение първо 7](#_Toc120821989)

[9. Реализация на условие … - допълнение второ 8](#_Toc120821990)

[10. Реализация на допълнение трето 8](#_Toc120821991)

[III. Упътване за употреба 9](#_Toc120821992)

[1. Впишете съответната част от проекта 9](#_Toc120821993)

[2. Впишете съответната част от проекта 9](#_Toc120821994)

[IV. Примерно действие на програмата 10](#_Toc120821995)

[1. Условие A 10](#_Toc120821996)

[2. Условие B 10](#_Toc120821997)

[3. Условие C 10](#_Toc120821998)

[4. Условие D 10](#_Toc120821999)

[5. Условие E 10](#_Toc120822000)

[6. Условие F 11](#_Toc120822001)

[7. Допълнение първо 11](#_Toc120822002)

[8. Допълнение второ 11](#_Toc120822003)

[9. Допълнение трето 11](#_Toc120822004)

Задание на проекта

**Информационна система „Студенти“**

Да се напише компютърна програма, реализираща информационна система, която поддържа студентска група (факултетен номер, ЕГН, три имена, списък с дисциплини, пол, възраст, статут-активен/прекъснал/завършил). Програмата съхранява и обработва данни за студентите в групата. Максималния брой студенти в една група е 30. Всеки студент има 5 оценки по дисциплините (име, оценка) от 1 семестър

**Базова задача**

1. Меню за избор на функциите от програмата.

**Функции от програмата са:**

1. Добавяне на студенти в група:

а. Добавяне на студенти в групата, чрез добавяне в масива, като не трябва да се превишава максималният брой студенти в една група (30).

1. Извеждане на всички студенти на екрана

а.Извеждане на всички студенти, чрез подходящо форматиране  в таблица

D. Търсене и извеждане на екрана:

а. Търсене и извеждане на студенти с оценка по БП в определен в интервал [min – max]

b. Търсене и извеждане на студенти с минимум един предмет, с оценка Слаб 2

E. Сортиране и извеждане на студенти на екран:

a. Сортиране на студентите по малко име по азбучен ред

F. Работа с външен файл (двоичен или текстов)

a. Извеждане на масива със студенти във файл

b. Въвеждане на масива със студенти от файл

**Допълнение първо (+ базова задача)**

G. Да се създаде подменю към основното с нови функции за:

а. Отделяне на студенти с определен статут (въвежда се от потребителят) и сортиране по факултетен номер.

b. Отделяне на студенти с успех в интервал (мин -макс) и сортиране по ЕГН.

**Допълнение второ (+ базова задача)**

H. Въвеждане/Актуализация на оценки и среден успех за студент

а. Добавяне в структурата на поле среден успех.

b. Въвеждане/Актуализацията на данни (оценки и среден успех), става по въведен факултетен номер на студент.

i. Ако студентът присъства в списъка(масива):

1. Проверява се статута му:

a. Активен – при въвеждане/актуализация на оценки, се изчислява автоматично средният успех. Всяка промяна на оценка изисква запитване за промяна! Ако няма оценка, тя се актуализира без запитване!

b. Прекъснал/завършил – НЕ могат да се актуализират оценки

c. Ако студента не присъства в списъка(масива), се извежда пояснително съобщение на екрана.

I. Смяна на статут на студент

a. Въвежда се факултетният номер на студента(ако го има, се въвежда и новият статут). Ако статутът е активен или прекъснал се сменя с новият. Ако статутът е завършил НЕ СЕ допуска промяна!

**Допълнение трето (+ базова задача)**

**J.** Данните в програмата да се попълват автоматично от файл при стартиране и да се записват автоматично във файл при затваряне на програмата.

Анализ на решението

Структура за данните в програмата

|  |  |  |
| --- | --- | --- |
| Структура | Обяснение | Примерени стойности |
| struct student\_info {  bool active;  long long int faculty\_number;  long long personal\_id\_number;  string full\_name;  string gender;  int age;  string status\_of\_student;  double sreden\_uspeh; | * аctive – Използва се за да се установи дали има данни въведени за пореден студент във функцията return\_num\_students като става true ако има въведени данни за студента * faculty\_number – съхранява факултетния номер * personal\_id\_number – съхранява ЕГН * full\_name – съхранява пълно име * gender – съхранява пол * status\_of\_student – съхранява статут на студента(graduated, dropout, active) * sreden\_uspeh – съхранява сметнат среден успех за студент | true  23621654  6101057509  Gilgamesh Aljabari  Male  18  active  4.50 |
| discipline disciplines[5] = {  {"Basis of programming“, 0}  {"Maths", 0},  {"Web Design", 0},  {"Electronics", 0},  {"Sport", 0}  };  }; | * discipline[5] – масив от тип discipline с 5 члена като има един за всеки предмет със име на предмета и оценка * name – променлива от помощната структура която съхранява името на дисциплината * grade съхранява оценката за дисциплината | 6  6  6  6  6 |

Реализация на условие A

Меню с избор на функциите от програмата

В алгоритъма се използва switch, case statement като от потребителя се въвежда опция от 0 до 9. Първоначално се извеждат функционалностите на програмата. При въвеждане на опция въведения вход се валидира и ако е се различава от очаквания на потребителя се пояснява, че трябва да бъде въведена валидна опция. Функцията int menu(student student [30] ) се извиква от главната функция int main().

(Ако за някое условие се използва повече от една функция, всяка една от тях се представя с блоксхема на алгоритъма; функция, с която е реализиран алгоритъма; входни данни на функцията; изходни данни или данни който се извеждат. Изтрийте този помощен текст.)

Блок схема на алгоритъмаКартина, която съдържа текст, диаграма, План, разписка

Описанието е генерирано автоматично

Функция, с която е реализиран алгоритъма

Int main(), като в нея се създава масив от структура тип student\_info и се извиква главното меню int menu()

#### Входни данни на функцията

Тук се представят входните данни на функцията: какви са, защо са избрани точно те, за какво ще се използват.

#### Изходни данни на функцията или данни, които се извеждат

Главната функция Int main() връща 0 при успешно изпълнение на програмата.

### **Функция, с която е реализиран алгоритъма**

int menu();

## 2.4.1 Входни данни на функцията

Въвежда се число от 0 до 9 като опциите на програмата са от 1 до 9, а 0 е резервирано за излизане от програмата.

2.4.2. Изходни данни на функцията

В зависимост от избраната има различни изходни данни. Опция 0, 1, 5, и 6 нямат изходни данни тъй като те са съответно функциите за въвеждане (собственоръчно или чрез файл), извеждане и изход от програмата

Реализация на условие B

Алгоритъм въвеждане на студенти в масива

Програмата пита потребителя колко студенти иска да въведе в масива. Ако въведе число по-малко от 1 или по-голямо от 30 се извежда грешка която казва на потребителя да въведе валиден брой студенти. Ако въведените досега студенти са по-малко от 30 на потребителя се позволява да въведе студенти, ако те са 30 се извежда пояснително съобщение което казва на потребителя, че не могат да се добавят повече студенти в масива.

Блок схема на алгоритъма

Тук се поставя блок схема на описания алгоритъм.

Функция, с която е реализиран алгоритъма

Алгоритъма е реализиран с функция int input()

#### Входни данни на функцията

Бройка студенти

#### Изходни данни на функцията или данни, които се извеждат

Функцията връща 0 при успешно изпълнение. Ако въведе число по-малко от 1 или по-голямо от 30 се извежда грешка която казва на потребителя да въведе валиден брой студенти. Ако са въведени максималния брой студенти на потребителя се извежда пояснително съобщение

Реализация на условие C

Алгоритъм извеждане на студенти от масива

Извеждат се всички досега въведени студенти. Ако няма въведени студенти досега се извежда пояснително съобщение, че няма въведени студенти в масива досега и няма какво да се извежда.

Блок схема на алгоритъма

Тук се поставя блок схема на описания алгоритъм.

Функция, с която е реализиран алгоритъма

Int output()

#### Входни данни на функцията

Функцията няма входни данни

#### Изходни данни на функцията или данни, които се извеждат

При успешно изпълнение функцията връща 0. Ако няма въведени студенти в масива при избиране на опцията се извежда пояснително съобщение, че няма въведени студенти в масива.

Реализация на условие D

Алгоритъм търсене и извеждане на екран

В алгоритъма има подменю което пита потребителя дали иска да търси и извежда студенти с оценка по БП в интервал или дали иска да търси студенти с оценка слаб 2.

Блок схема на алгоритъма

Тук се поставя блок схема на описания алгоритъм.

Функция с която е реализиран алгоритъма

Алгоритъма се реализира чрез функция int search\_in\_range\_or\_look\_for\_fail()

#### Входни данни на функцията

Опция 1 или 2. При избор на всякакво друго число се извежда съобщение грешка. Ако се избере опция 1 се въвежда долната граница на интервала и горната граница. Долната граница не може да е по-малка от ,а горната не може да е по-голяма от 6. Ако потребителя въведе по-малко от 2 или по-голямо от 6 се извежда пояснително съобщение. Ако се избере опция 2 се извеждат студенти с минимум една оценка Слаб 2. Ако няма студенти с оценка Слаб 2 извежда съобщение поясняващо това.

#### Изходни данни на функцията или данни, които се извеждат

При опция 1 се извеждат имената на студенти с оценка по БП в интервала, а в опция 2 се извеждат имената на студенти, който имат минимум една оценка Слаб 2.

Реализация на условие E

Сортиране и извеждане на студенти на екран

В алгоритъма се сортират студенти по малко име и се извеждат целите имена на студенти.

Блок схема на алгоритъма

Тук се поставя блок схема на описания алгоритъм.

Функция с която е реализиран алгоритъма

Алгоритъма се реализира с функция sort\_by\_first\_name()

#### Входни данни на функцията

Функцията няма входни данни. За да се стартира се избира опция 4.

#### Изходни данни на функцията или данни, които се извеждат

Извеждат се съобщение “Sorted Students by First Name” и се извеждат студентите сортирани по малко име.

Реализация на условие F

Работа с външен файл

Масива от студенти се записва във външен файл и се въвежда(чете) от външен файл

Блок схема на алгоритъма

Тук се поставя блок схема на описания алгоритъм.

Функция с която е реализиран алгоритъма

Алгоритъма е реализиран с две функции int write\_to\_file() и функция int read\_array\_from\_file()

#### Входни данни на функцията

Няма входни данни. Функцията за извеждане във файл се избира със опция 5 и функцията за четене от файл се избира с опция 6.

#### Изходни данни на функцията или данни, които се извеждат

Пояснително съобщение което казва, че четенето или записването във файла е успешно.

Реализация на условие G - допълнение първо

Създаване на подменю към основното с функции за отделяне на студенти с определен статут и сортиране студентите с този статут и отделяне на студенти с успех в интервал и сортиране по ЕГН

Създадена е функция в която се съдържа меню за избор между функцията за отделяне на студенти по статут и отделяне на студенти с успех в интервал. Ако се избере невалидна опция се извежда съобщение грешка и потребителят трябва да въведе опция пак.

Блок схема на алгоритъма

Тук се поставя блок схема на описания алгоритъм.

Функция с която е реализиран алгоритъма

Алгоритъма е реализиран с функция int menu\_for\_function\_seperate\_students\_by\_grades\_or\_status(), като в нея се съдържа меню за избор на една от функциите separate\_students\_by\_status() и separate\_students\_by\_grades()

#### Входни данни на функцията

Въвежда се опция 1 за отделяне на студенти по статут и опция 2 за отделяне на студенти с успех в интервал.

#### Изходни данни на функцията или данни, които се извеждат

Извеждат се отделените студенти по статут или оценки. Ако се въведе невалидна опция( различна от 1 или 2) се извежда пояснително съобщение, че е въведена невалидна опция.

Реализация на условие H - допълнение второ

Въвеждане/Актуализация на оценки и среден успех за студент

Добавена е в структурата променлива среден успех като тя се смята чрез функция calculate\_average\_grade(). Функцията позволява актуализация на данни(оценки и среден успех), съответно средния успех се смята наново. Актуализацията става по въведен факултетен номер като програмата търси съвпадение. Ако няма съвпадение се извежда съобщение гласящо, че студента го няма в базата данни. Ако няма въведени студенти се извежда съобщение, че няма въведени студенти. Проверява се статута на студента и ако е различен от “active” (“graduated” или “dropout”) се извежда съобщение което казва на потребителя, че не могат да се актуализират оценките за този студент.

Блок схема на алгоритъма

Тук се поставя блок схема на описания алгоритъм.

Функция с която е реализиран алгоритъма

Използва се функция change\_data() и помощна функция return\_num\_students() която връща досега въведения брой студенти. След промяна оценките се смята средния успех наново. Използва се и функция calculate\_average\_grade за смятане наново средния успех и също така convertToLowerCase() за изпълняване на програмата правилно дори ако отговорите “yes” или “no” са въведени с големи букви.

#### Входни данни на функцията

Въвежда се факултен номер на студент. След това за всяка промяна в оценка потребителя се пита дали иска да направи промяна за нея. Ако избере “yes” тогава му се дава опцията да въведе оценката за съответния предмет.

#### Изходни данни на функцията или данни, които се извеждат

Програмата връща 0 при правилно изпълнение, а при различните видове грешки връща пояснителен текст. При въвеждането също се извежда пояснителен текст какво трябва да се въведе.

Реализация на условие I

Смяна статут на студент

Въвежда се факултетен номер. Ако се намери студент с въведения факултетен номер се въвежда и новия статут. При текущи статуи активен или прекъснал се позволява промяна, а при текущ статут завършил не се позволява промяна.

Блок схема на алгоритъма

Тук се поставя блок схема на описания алгоритъм.

Функция с която е реализиран алгоритъма

Използва се функция change\_status\_of\_student() и се изполват помощни функции return\_num\_students, и convertToLowerCase()

#### Входни данни на функцията

Въвежда се факултетен номер, ако статута на студента е “active” или “dropout” се въвежда нов статут на студент.

#### Изходни данни на функцията или данни, които се извеждат

Извеждат се съобщения за грешки и пояснителни съобщения за въвеждането.

Реализация на условие J - допълнение трето

Попълване на данни автоматично от файл и записване във файл при затваряне на програмата

Използват

Блок схема на алгоритъма

Тук се поставя блок схема на описания алгоритъм.

Функция с която е реализиран алгоритъма

Алгоритъма е реализиран с две функции int write\_to\_file() и функция int read\_array\_from\_file() като те са поставени на подходящите места.

#### Входни данни на функцията

Функцията чете автоматично данни за студентите от файл при стартиране на програмата.

#### Изходни данни на функцията или данни, които се извеждат

Извеждат се пояснителни съобщение, че данните са съответно успешно въведени от файла при стартиране на програмата и, че са записани при затваряне.

Упътване за употреба

Представете какво трябва да се въведе за правилното използване на програмата, какво се извежда в резултат на нея. Какви са очакваните резултати и какво ще се случи, ако се въведат невалидни данни. Изтрийте този текст, след като се запознаете с него.

Впишете съответната част от проекта

Очаквани входни данни

Моля, опишете.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

Моля, опишете.

Впишете съответната част от проекта

Очаквани входни данни

Моля, опишете.

Очакван резултат от изпълнението на конкретния фрагмент от проекта

Моля, опишете.

Примерно действие на програмата

Условие A

Снимка на изгледа с примерни входни данни

Моля, поставете снимката тук.

Снимка на изгледа с примерни изходни данни

Моля, поставете снимката тук.

Условие B

Снимка на изгледа с примерни входни данни

Моля, поставете снимката тук.

Снимка на изгледа с примерни изходни данни

Моля, поставете снимката тук.

Условие C

Снимка на изгледа с примерни входни данни

Моля, поставете снимката тук.

Снимка на изгледа с примерни изходни данни

Моля, поставете снимката тук.

Условие D

Снимка на изгледа с примерни входни данни

Моля, поставете снимката тук.

Снимка на изгледа с примерни изходни данни

Моля, поставете снимката тук.

Условие E

Снимка на изгледа с примерни входни данни

Моля, поставете снимката тук.

Снимка на изгледа с примерни изходни данни

Моля, поставете снимката тук.

Условие F

Снимка на изгледа с примерни входни данни

Моля, поставете снимката тук.

Снимка на изгледа с примерни изходни данни

Моля, поставете снимката тук.

Допълнение първо

Снимка на изгледа с примерни входни данни

Моля, поставете снимката тук.

Снимка на изгледа с примерни изходни данни

Моля, поставете снимката тук.

Допълнение второ

Снимка на изгледа с примерни входни данни

Моля, поставете снимката тук.

Снимка на изгледа с примерни изходни данни

Моля, поставете снимката тук.

Допълнение трето

Снимка на изгледа с примерни входни данни

Моля, поставете снимката тук.

Снимка на изгледа с примерни изходни данни

Моля, поставете снимката тук.