Технически Унивеситет – Варна

Курсов Проект

Информационна система за обслужване на студенти

(ООП – 1ва част)

Изготвил : Николай Николаев Панев

Фак. Номер: 20621511

Факултет : ФИТА

Специалност : СИТ

Глава 1

Задание

# Работа с командния ред

Вашата програма трябва да позволява на потребителя да отваря файлове (open), да извършва върху тях някакви операции, след което да записва промените обратно в същия файл (save) или в друг, който потребителят посочи (save as). Трябва да има и опция за затваряне на файла, без записване на промените (close). За целта, когато програмата ви се стартира, тя трябва да позволява на потребителя да въвежда команди и след това да ги изпълнява.

Когато отворите даден файл, неговото съдържание трябва да се зареди в паметта, след което файлът се затваря. Всички промени, които потребителят направи след това трябва да се пазят в паметта, но не трябва да се записват обратно, освен ако потребителят изрично не yкаже това.

Във всеки от проектите има посочен конкретен файлов формат, с който приложението ви трябва да работи. Това означава, че:

1. то трябва да може да чете произволен валиден файл от въпросния формат;
2. когато записва данните, то трябва да създава валидни файлове във въпросния формат.

Както казахме по-горе, потребителят трябва да може да въвежда команди, чрез които да посочва какво трябва да се направи. Командите могат да имат нула, един или повече параметри, които се изреждат един след друг, разделени с интервали.

Освен ако не е казано друго, всяка от командите извежда съобщение, от което да е ясно дали е успяла и какво е било направено.

Дадените по-долу команди трябва да се поддържат от всеки от проектите. Под всяка от тях е даден пример за нейната работа:

### Open

Зарежда съдържанието на даден файл. Ако такъв не съществува се създава нов с празно съдържание.

Всички останали команди могат да се изпълняват само ако има успешно зареден файл.

След като файлът бъде отворен и се прочете, той се затваря и приложението ви вече не трябва да работи с негo, освен ако потребителят не поиска да запише обратно направените промени (вижте командата save по-долу), в който случай файлът трябва да се отвори наново. За целта трябва да изберете подходящо представяне на информацията от файла.

Ако при зареждането на данните, приложението ви открие грешка, то трябва да изведе подходящо съобщение за грешка и да прекрати своето изпълнение.

* open C:\Temp\file.xml Successfully opened file.xml

### Close

Затваря текущо отворения документ. Затварянето изчиства текущо заредената информация и след това програмата не може да изпълнява други команди, освен отваряне на файл (Open).

* close

Successfully closed file.xml

### Save

Записва направените промени обратно в същия файл, от който са били прочетени данните.

* save

Successfully saved file.xml

### Save As

Записва направените промени във файл, като позволява на потребителя да укаже неговия път.

* saveas "C:\Temp\another file.xml" Successfully saved another file.xml

### Help

Извежда кратка информация за поддържаните от програмата команди.

* help

The following commands are supported:

open <file> opens <file>

close closes currently opened file

save saves the currently open file

saveas <file> saves the currently open file in <file> help prints this information

exit exists the program

### Exit

Излиза от програмата

* exit

Exiting the program...

# Студент

Да се напише програма, реализираща информационна система за обслужване на студенти. Програмата съхранява и обработва необходимите данни във файл.

За всеки студент се съхранява следната информация:

* Име — символен низ с произволна дължина
* Факултетен номер
* Текущо записан курс, специалност, група
* статус (записан, прекъснал, завършил)
* Среден успех от следването до момента

Учебните дисциплини се характеризират с име и тип (задължителни / избираема). За всяка специалност има списък от дисциплини и курс, в който могат да бъдат записвани, като за простота можете да приемете, че всяка дисциплина може да се записва само в рамките на точно един курс.

За всеки студент се пазят всички записани дисциплини и оценки по тези от тях, по които са положени изпити.

* добавяне на оценки/изпити (за даден предмет, оценка и студент)
* записване на задължителни/избираеми предмети
* отпечатване на протоколи
* академична справка за оценките на даден студент (списък с всички взети изпити

,съответните оценки и успех на студента). Да се включи също и списък с невзетите изпити (предмети, които са записани, но за тях няма оценки, те да се включат в пресмятането на средния успех на студента като двойки), ако има такива.

След като приложението отвори даден файл, то трябва да може да извършва посочените по-долу операции, в допълнение на общите операции (open, close, save, save as, help и exit):

|  |  |
| --- | --- |
| enroll <fn> | записване на студент с име <name> в 1 курс на специалност <program> |
| <program>  <group> | в група <group> и с факултетен номер <fn>. |
| <name> |  |
| advance <fn> | записва студент в следващ курс |
| change <fn>  <option>  <value> | <option> е едно от program, group, year.  Прехвърля студента с факултетен номер <fn> в нова специалност (program), група (group) или курс (year) зададени чрез <value>.  Прехвърлянето в група е възможно винаги. |

|  |  |
| --- | --- |
|  | Прехвърлянето в следващ курс е възможно само, ако студентът е положил успешно изпитите по всички задължителни предмети от минали курсове, с евентуално изключение на максимум два курса. Прехвърляне в курс, различен от следващия, не е възможно.  Прехвърлянето в друга специалност е възможно само, ако студентът е положил успешно изпити по всички задължителни предмети от минали курсове на новата специалност. |
| graduate  <fn> | Отбелязва студента като завършил, но само ако е положил успешно изпити по всички записани предмети. |
| interrupt  <fn> | Маркира студента с факултетен номер <fn> като прекъснал. Прекъсналите студенти не могат да се явяват на изпити, да записват учебни дисциплини или да сменят специалност, група или курс. |
| resume  <fn> | Възстановява студентските права на студента с факултетен номер  <fn>. |
| print <fn> | Извежда справка за студента с факултетен номер <fn>. |
| printall  <program>  <year> | Извежда справка за всички студенти в дадена специалност и курс. |
| enrollin  <fn>  <course> | Записва студента с факултетен номер <fn> в дисциплината с име  <course>. Записването е позволено само на дисциплини от съответния курс и специалност. |
| addgrade  <fn>  <course>  <grade> | Добавя оценка <grade> по дисциплината <course> на студента с факултетен номер <fn>. Явяването на изпит е позволено само за дисциплини, които са записани. |
| protocol  <course> | Извежда протоколи за всички студенти, записани в дадена дисциплина  <course>. За всяка специалност и курс се извежда отделен протокол. Студентите в протокола са подредени по факултетен номер в нарастващ ред. |
| report <fn> | Извежда академична справка за оценките на даден студент (списък с всички взети изпити, съответните оценки и успех на студента). Да се включи също и списък с невзетите изпити (дисциплини, които са записани, но за тях няма оценки, те да се включат в пресмятането на средния успех на студента като двойки), ако има такива. |

**Бонуси:**

* Да се поддържат дисциплини, които могат да бъдат записвани в няколко възможни курса (например 2, 3 и 4).
* Да се поддържат кредити за всяка избираема дисциплина и минимален брой за специалността и да се извежда справка колко още кредита от избираеми дисциплини трябва да вземе студента, за да може да се дипломира (кредитите от избираеми дисциплини, които студента не е взел, не се включват в текущите кредити).

Глава 2

Кратко описание

на програмата

Предназначение

Програмата е създадена за обслужване на студенти, давайки възможност за оценяване на студент, записването му в университета, преминаване в следваща година, завършване и други. Програмата също помага за създаването на нови специалности и дисциплини към тях.

Начин на работа

Програмата работи в командният ред, т.е. за нейното използване се въвеждат команди от потребителя в конзолата. Всяка команда има собствен специфичен синтаксис, който потребителя може да види след въвеждане на команда „help“(помощ). На екранът ще се изведат всички достъпни до потребителя команди, техният синтаксис и кратко описание на това, което правят (фиг.1)

Text

Description automatically generated фиг.1

Възможности за развитие на програмата

Понеже в текущата си версия програмата не може да работи с файлове, и използва само предварително зададени и временни данни, в бъдеще може да бъде имплементирана работа с XML файлове, локални и глобални бази от данни и дори да бъде създаден стандарт за текстов файлов формат за работа с програмата. Програмата също може да бъде реализирана като онлайн приложение(или онлайн портал) за обслужване на студенти от един университет, град, или дори от една държава.

Глава 3

Проектиране

Обща структура на проекта

Проекта се състои от няколко „структурни“ класа:

* Person – Абстрактен клас, съдържащ в себе си общите даннови полета и методи на човек, за когото могат да бъдат създадени подкласове(Студент, Преподавател, Квестор и др.)
  + Student – Всяка инстанция представлява нов студент в програмата, всеки със свой факултетен номер, специалност, група и име. (фиг.2)

A picture containing diagram

Description automatically generated фиг.2

* Major –Всяка инстанция представлява нова специалност, всяка със собствено име и дисциплини за 4те години на обучение(фиг.3)

Text

Description automatically generatedфиг.3

* Discipline – Всяка инстанция представлява нова дисциплина, всяка със собствено име и булева стойност дали е задължителна или не. Всяка дисциплина може да бъде добавена във всяка специалност, ако вече не съществува в тази специалност. (фиг.4)

Text

Description automatically generated with low confidence фиг.4

* Grade – Всяка инстанция представлява нова оценка, всяка със собствена дисциплина и стойност между 2.00 и 6.00. Класът за студент съдържа метод за оценяване, който добавя оценка в лист с оценки, съдържащ се в класа студент.

Text

Description automatically generated

Един абстрактен клас съдържащ методите за работа в командния ред:

* Menu – Съдържа абстрактният метод за работа с командният ред, както и всички частни методи използвани в него.

Text

Description automatically generated

И основнят клас на програмата, съдържащ метод main(String[] args).