

Задачи

Задача 1

Да се създаде структура **Student**, със следните полета:

- mId – идентификационен номер
- mFirstName - име
- mLastName - фамилия
- mFacultyNumber – факултетен номер
- mAge - възраст
- mSpecialty – специалност
- mFaculty - факултет

Да се реализират следните функции:

- **initStudent(Student& student)** – инициализира всички полета на student
- **clearDynamic(Student& student)** – изчиства динамичната памет на student
- **writeStudents(char* filename, Student* students, size_t numStudents)** – записва учениците в бинарен файл с име filename
- **Student* readStudents(char* filename, size_t& numReadStudents)** – прочита ученици, записани в двоичен файл с име filename и задава numReadStudents с броя на прочетените ученици от файла. Накрая връща масив с прочетените ученици
- **printStudents(Student* students, size_t numStudents)** – принтира на екрана информацията за учениците

Задача 2

Да се създаде структура, с която да се представя **времето** в следния **формат**:

dd MMM YYYY, hh:mm

Пример: 28 Feb 2021, 14:22

За целта първо ще ни трябва структура **Time** със следните полета:

- mHours
- mMinutes

После и структурата **Date** с полета:

- mYear
- mMonth
- mDay
- mTime

Да се реализират следните функции:

- **Date readDate(istream& in)** – прочита датата от поток и я връща
- **void writeDate(ofstream& out)** – записва датата в потока

Задача 3

Роси и Ванката били много заети хора. Постоянно имали нужда от бърз начин да си запишат някаква бележка и да имат възможността да я променят и прочетат, когато си искат. Общо взето трябвало им StickyNotes, обаче тъй като са супер яките програмисти решили да си накодят тяхна версия. Нека помогнем на Роси и Ванката, като им напишем програма за note tracking. За целта ще представим всяка бележка със следните данни:

- дата на създаване
- съдържание
- дата на последна промяна

Hint: за датите може да използвате структурата и функциите от миналата задача

Ще ни трябват и следните функции:

- **Note* readNotes(char* filename, size_t& numReadNotes)** – прочита бележки записани във файл filename, и записва в numReadNotes броя на прочетените бележки. Връща масив от бележки
- **printNotes(const Note* notes)** – принтира цялата информация за бележките
- **editNote(Note* notes, size_t index)** – първо принтира информацията за бележките и после дава на потребителя да запише ново съдържание на бележката на място index
- **saveNotes(char* filename, Note* notes, const size_t& numNotes)** – записва в двойчен файл всички бележки в масива notes