

Отчёт по проекту "База Данных Абитуриентов МехМата"

Проект заключается в том, чтобы создать базу данных абитуриентов МехМата на специалитет. База данных содержит такие поля, как id, Фамилия, Имя, Отчество, дата рождения, Регион, дата окончания школы(гимназии, лицея), 4 поля для предметов, 4 поля для баллов, средний балл за эти предметы соответственно. Создание базы данных проходит поэтапно.

I этап: На первых порах реализован генератор случайных ФИО, регионов(только номера), дат рождения, лет окончания школы, предметов, сданных для того, чтобы поступить на МехМат и баллов, полученных на ЕГЭ по этим предметам. Также реализован класс fStudent(от future Student), в полях которого находятся номер идентификатор id, 3 вектора строк для фамилии(fam_name), имени(name), отчества(mid_name), номер региона(region) и вектор строк сданных предметов(object).

В классе fStudent реализованы методы: конструктор, печать и "редактор записи".

Списки начальных фамилий, имён, отчеств и предметов ЕГЭ, которые ученики сдали, чтобы поступить на МехМат, хранятся соответственно в файлах family_names.txt, names.txt, mid_names.txt, objects.txt. Выгрузка данных происходит в файл outfile.txt. На этом первый этап проекта завершён.

II этап: Реализован класс базы данных DB в файлах "Database.h" и "Database.cpp". В данном классе имеются такие функции, как loadDB, printDB, saveDB, add, select, getFreeID, которые отвечают соответственно за загрузку базы данных из файла, который сделал генератор, вывод базы данных, сохранение базы данных (в файл, отличный от основного файла), добавление в базу данных некоторых данных, выборка и генерация свободного, нового ID.

Реализован оператор копирования в классе fStudent. Добавление проходит в 2 этапа: а) проверка корректности данных, выделение памяти под новую запись, затем добавление этой записи в множество базы данных set.

Выборка проходит так: мы вводим критерии, по которым хотим найти запись в базе данных, последним параметром является би-

товая маска, которая определяет, по какому(-им) критерию(-иям) будет проходить выборка. Более подробно о параметрах описано в файле Database.cpp. Вывод проходит в файл dataout.txt. На этом второй этап проекта завершён.

III этап: Реализован файл "Parsing.cpp" по работе с языком запросов. Теперь при запуске программы в командной строке появляется список команд, которые мы можем выполнять над базой данных. Среди таких команд: "Exit"; "Generate"; "Load"; "Print"; "Save"; "Add"; "Select".

При вводе команды **"Exit"** мы завершаем работу с программой.

При вводе команды **"Generate"** через пробел ставим число - это количество элементов в нашей базе, которые генератор сгенерирует. Всё сгенерированное добро находится в файле "outfile.txt".

При вводе команды **"Load"** мы делаем так, что все записи из файла "outfile.txt" считываются в базу данных.

При вводе команды **"Print"** мы выводим всё содержимое нашей базы данных в файл "dataout.txt".

Команда "Save" помогает нам сохранить нашу базу данных в любой момент времени, сохранив её в файл "save.txt".

Команда "Add" позволяет добавить в базу данных свою запись. О том, как это сделать, описано в списке команд, который появляется при запуске программы.

Команда "Select" позволяет выбрать в базе данных записи по определённым критериям. О том, как это сделать, описано в списке команд, который появляется при запуске программы.

На этом третий этап проекта завершён.