Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Дисциплина: Программируемые мобильные системы

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе

на тему

**РАЗРАБОТКА ПРОГРАММЫ УЧЕТА ЗАРАБОТНОЙ ПЛАТЫ СОТРУДНИКОВ ПРЕДПРИЯТИЯ**

Выполнил: студент группы

Ф.И.О.

Проверил: Ф.И.О. преподавателя

Минск 2022

Содержание

1. Требования к программе

2. Конструирование программы

2.1 Разработка модульной структуры программы

2.2 Выбор способа организации данных

2.3 Разработка перечня пользовательских функций программы

Требования к программе

Необходимо разработать программу расчета заработной платы сотрудников предприятия.

Сведения о сотрудниках предприятия содержат:

- Ф.И.О. сотрудника;

- табельный номер;

- год;

- месяц;

- количество проработанных часов за месяц;

- почасовой тариф.

Рабочее время свыше 144 часов считается сверхурочным и считается в двойном размере.

Индивидуальное задание: рассчитать размер заработной платы каждого сотрудника за вычетом подоходного налога, который составляет 12%. Определить объем выплат конкретному сотруднику за требуемый период времени, который вводится с клавиатуры.

Также необходимо реализовать авторизацию для входа в систему, функционал администратора и пользователя.

Исходные данные

1. Тема: разработка программы расчета заработной платы сотрудников предприятия
2. Язык программирования: C++
3. Среда разработки: Microsoft Visual Studio
4. Вид приложения: консольное
5. Парадигма программирования: процедурная
6. Способ организации данных: структуры
7. Способ хранения данных: файлы
8. Каждая логически завершенная подзадача должна быть реализована в виде отдельной функции
9. Построение программного кода должно соответствовать соглашению о коде «C++ Code Convention»
10. К защите курсовой работы представляются: консольное приложение и пояснительная записка
11. Текст пояснительной записки оформляется в соответствии со стандартом предприятия СТП 01-2017

Функциональные требования

Первый этап работы программы – авторизация – предоставление прав доступа.

В рамках данного этапа считываются данные из файла с учетными записями пользователей следующего вида:

- login;

- hashPassword;

- role;

- access.

Поле login хранит логин пользователя, hashPassword хранит зашифрованный пароль пользователя, role хранит права доступа, где 1 – администратор, 0 – пользователь, access хранит разрешение на вход в систему, где 1 – доступ разрешен, 0 – доступ запрещен.

Если файла с учетными записями пользователей не существует, то создается файл, где записываются данные администратора.

По соображениям безопасности, при вводе пароля на экране выводятся «звездочки».

Регистрация новых пользователей осуществляется администратором в режиме работы с учетными записями пользователе. При регистрации выполняется проверка на уникальность логина.

Также возможна регистрация пользователей при входе, после которой администратор должен подтвердить регистрацию в режиме редактирования учетных записей пользователей, путем изменения поля access.

Второй этап работы программы – работа с данными, которая становится доступной только после успешного входа в систему. Данные о сотрудниках хранятся в файле и имеют структуру:

- уникальный ID записи;

- фамилия сотрудника;

- имя сотрудника;

- отчество сотрудника;

- табельный номер;

- год;

- месяц;

- количество проработанных часов за месяц;

- почасовой тариф.

Для работы с данными предусмотрены два функциональных модуля: модуль администратора и модуль пользователя.

Модуль администратора содержит следующие подмодули:

1. Управление учетными записями пользователей:

- просмотр всех учетных записей пользователей

- добавление новой учетной записи пользователя

- редактирование учетной записи пользователя

- удаление учетной записи пользователя

Так как существует возможность регистрации при входе, администратор подтверждает аккаунт через подмодуль «Редактирование учетной записи пользователя».

Допускается создавать учетную запись пользователя для нескольких администраторов.

Администратору запрещено удалять или изменять права доступа и разрешение на вход в систему самому себе.

1. Работа с данными:
   1. Режим редактирования:

- просмотр всех записей;

- добавление новой записи;

- удаление записи;

- редактирование записи;

* 1. Режим обработки данных:

- выполнение индивидуального задания в рамках данной курсовой работы;

- поиск данных;

- сортировка;

Модуль пользователя содержит следующие подмодули:

1. Просмотр всех данных;
2. Режим обработки данных:

- выполнение индивидуального задания в рамках данной курсовой работы;

- поиск данных;

- сортировка;

Для реализации перечисленных модулей и подмодулей созданы меню с соответствующими пунктами.

В данной курсовой работе предусмотрены:

1. Исключительные ситуации:

- отсутствие файла с учетными записями пользователей;

- проверка на уникальность логина при регистрации и редактировании учетной записи пользователя;

- запрет на изменение некоторых полей и удаление учетной записи пользователя;

- введенные пользователем данные не соответствуют формату поля;

- введенные пользователем данные нелогичны (например, почасовой тариф меньше либо равен 0);

- файл с данными для чтения не существует;

- номер удаляемой записи выходит за пределы массива;

1. Возможность перемещения по программе;
2. Запрос на удаление при необратимых действиях: подтверждение при удалении данных;
3. Обратную связь с пользователем через сообщения.

Требования к программной реализации

1. Все переменные и константы должны иметь осмысленные имена в рамках тематики данной курсовой работы. Переменным рекомендуется присваивать имена, состоящие из букв нижнего регистра, для формирования составного имени используется «верблюжья нотация» (например, flagExit). Константам присваиваются имена, состоящие из букв верхнего регистра, и используется нижнее подчеркивание (например, FILE\_OF\_ACCOUNTS);
2. Имена функции должны быть осмысленными, начинаются с буквы нижнего регистра и строятся по принципу глагол+существительное (например, addAccount). Если функция выполняет проверку и возвращает результат типа bool, то ее название должно начинаться с глагола is (например, isNumberNumeric);
3. Не допускается использование оператора прерывания goto
4. Код не должен содержать неименованных числовых констант, неименованных строковых констант. Подобного рода информацию следует представлять как глобальные константы. По правилам качественного стиля программирования тексты всех информационных сообщений, выводимых пользователю в ответ на его действия, также оформляются как константы.
5. Код необходимо комментировать (как минимум в части объявления струк- тур, массивов/векторов, прототипов функций, нетривиальной логики).
6. Код не должен дублироваться – для этого существуют функции!
7. Одна функция решает только одну задачу (например, не допускается в одной функции считывать данные из файла и выводить их на консоль – это две разные функции). При этом внутри функции возможен вызов других функций.
8. Выполнение операций чтения/записи в файл должно быть сведено к минимуму (т. е. после однократной выгрузки данных из файла в массив дальнейшая работа ведется с этим массивом, а не происходит многократное считывание данных из файла в каждой функции).
9. Следует избегать глубокой вложенности условных и циклических кон-струкций: вложенность блоков должна быть не более трех.
10. Следует избегать длинных функций: текст функции должен умещаться на один экран (размер текста не должен превышать 25–50 строк).
11. Следует выносить код логически независимых модулей в отдельные .cpp файлы и подключать их с помощью заголовочных .h файлов.

КОНСТРУИРОВАНИЕ

Разработка модульной структуры

Выбор способа организации данных

В качестве способа организации данных используются структуры следующего вида:

- Структура учётной записи пользователя:

- логин;

- зашифрованный пароль;

- права доступа;

- возможность входа в систему;

- Структура записи сотрудника:

- уникальный ID записи;

- фамилия сотрудника;

- имя сотрудника;

- отчество сотрудника;

- табельный номер;

- год;

- месяц;

- количество проработанных часов за месяц;

- почасовой тариф.

В качестве способа объединения входных данных используются глобальные динамические массивы структур.

Разработка перечня пользовательских функций