**ТЕХНИЧЕСКОЕ ЗАДАНИЕ ПО ПРОЕКТУ**

СУПЕРМАРКЕТ v1.0

Севкович К.А.

гр. 653501

Отдел кадров

Программа Отдел кадров предназначена для автоматического управления списком сотрудников какой-либо организации. Она является удобным инструментом в осуществлении контроля над необходимой информацией о работнике. Программа позволяет записывать и хранить данные о сотрудниках в базе данных, а так же удалять эту информацию при необходимости. Можно произвести поиск сотрудников по любой доступной программе информации, а так же редактировать уже существующую. Наиболее полезным является автоматический подсчёт времени, которое проработал сотрудник и автоматический подсчёт положенного ему отпуска

Программа рассчитана на пользователя, владеющего английским языком и имеющего минимальный уровень компьютерной грамотности.

Структуры данных

В программе использованы следующие структуры данных:

1. Для описания сотрудника используется следующая структура:

struct Employee

{

std::string Name;

Date BirthDate;

std::string PassportNumber;

std::string SocialSecurityNumber;

std::string Education;

std::string Position;

double Salary;

Date StartWork;

std::vector<std::string> Notes;

bool AtVacationStatus;

void SetName()

void SetBirthDate()

void SetPassportNumber()

void SetSocialSecurityNumber()

void SetEducation()

void SetPosition()

void SetSalary()

void SetStartWork()

void SetNotes()

void SetStatus()

};

1. Для хранения загруженных из базы данных файлов используется класс списка:

class ListEmployee

{

protected:

struct Node

{

Employee value;

Node\* next;

Node\* prev;

Node() { }

};

void deleteNode(Node\* &pNode);

Node\* head;

unsigned size;

public:

ListEmployee()

{

head = nullptr;

size = 0;

}

~ListEmployee();

void Add(Employee pNewEmployee);

void Remove(std::string pName, std::string pSocialSecurityNumber = "");

std::vector<int> FindByName(std::string pName = "");

std::vector<int> FindByBirthDate();

std::vector<int> FindByPassportNumber();

std::vector<int> FindBySocialSecurityNumber();

std::vector<int> FindByEducation();

std::vector<int> FindByPosition();

std::vector<int> FindBySalary();

std::vector<int> FindByStartWork();

std::vector<int> FindByStatus();

std::vector<int> FindByAtVacationStatus();

Employee GetAt(int index);

int GetSize();

bool IsEmpty();

};

1. Для работы с датами используется класс Date:

class Date

{

private:

int year;

int month;

int day;

void set(unsigned pYear, unsigned pMonth, unsigned pDay);

public:

Date()

{

this->year = this->month = this->day = 0;

}

// pDate should be in form "mm/dd/yyyy"

Date(std::string pDate);

Date(unsigned pYear, unsigned pMonth, unsigned pDay);

static Date GetCurrentDate();

void Set(unsigned pYear, unsigned pMonth, unsigned pDay);

void Set(std::string pDate);

std::string GetAsString() const;

int GetJulianDate() const;

int GetDateDifference(Date pDateSecond) const;

static bool IsDateValid(unsigned pYear, unsigned pMonth, unsigned pDay);

static bool IsLeapYear(unsigned pYear);

static std::vector<std::string> SplitString(std::string pString, char delim);

static bool IsDateStringValid(std::string pDate);

bool operator == (const Date &other) const;

int operator - (const Date &other) const;

};

Архитектура системы

Работа с программой организована в виде навигации по различным пунктам меню. Доступ к пунктам меню осуществляется нажатием на клавиатуре клавиш цифр, соответствующих позиции в меню. При выборе пункта меню, программа ведет пользователя далее с помощью интуитивно понятного интерфейса, который предусматривает обработку корректных и некорректных действий пользователя.