Министерство образования и науки РФ  
Федеральное государственное автономное  
образовательное учреждение высшего профессионального образования  
«НИТУ МИСиС»

Институт ИТКН

**Курсовая работа**

По дисциплине«Технология программирования»

Тема: «Мониторинг торговых точек. Контроль за остатками товара в торговых точках. Мониторинг цен на товар в торговых точках.»

Выполнил:  
студент гр. БПИ-20-2  
Бобров М. А.

Проверил:   
преп. Широков А. И.

Москва, 2021 г.

**Содержание**

[Техническое задание 3](#_Toc88775475)

[1. Наименование программы 3](#_Toc88775476)

[2. Область применения 3](#_Toc88775477)

[3. Основание для разработки 3](#_Toc88775478)

[4. Назначение разработки 3](#_Toc88775479)

[5. Технические требования к программе или программному изделию 3](#_Toc88775480)

[6. Технико-экономические показатели 4](#_Toc88775481)

[7. Стадии и этапы разработки 4](#_Toc88775482)

[7.1 Стадии разработки 4](#_Toc88775483)

[7.2 Этапы разработки 4](#_Toc88775484)

[8. Порядок контроля и приемки 4](#_Toc88775485)

[Эскизный проект 5](#_Toc88775486)

[1. Предварительная разработка структуры входных 5](#_Toc88775487)

[и выходных данных 5](#_Toc88775488)

[2. Уточнение методов решения задачи 6](#_Toc88775489)

[3. Разработка общего описания алгоритма решения задачи 6](#_Toc88775490)

[Технический проект 8](#_Toc88775491)

[1. Уточнение структуры входных и выходных данных 8](#_Toc88775492)

[2. Разработка алгоритма решения задачи 9](#_Toc88775493)

[3. Определение формы представления входных и выходных данных 11](#_Toc88775494)

[4. Определение семантики и синтаксиса языка 11](#_Toc88775495)

[5. Разработка структуры программы 11](#_Toc88775496)

[6. Окончательное определение конфигурации технических средств 13](#_Toc88775497)

Техническое задание

1. Наименование программы

Наименование программы – “Shop Keeper”. **НАПИСАТЬ БОЛЬШЕ ТЕКСТА В ТЗ, УБРАТЬ ТАБУЛЯЦИИ, ЗАМЕНИТЬ НА ОТСТУП В ФОРМАТИРОВАНИИ ТЕКСТА!!!!**

2. Область применения

Программа предназначена для контроля за остатками товара в торговых точках и для мониторинга цен товара в торговых точках. Учет ведется по одному типу товара.

3. Основание для разработки

Основанием для разработки является задание для курсовой работы. Организацией, утвердившей задание, является университет НИТУ «МИСиС».

4. Назначение разработки

Продукт разрабатывается в качестве вспомогательной программы для бизнеса. Функциональным и эксплуатационным назначением является использование её обученными работниками магазина.

5. Технические требования к программе или программному изделию

Программа должна обеспечивать выполнение следующих функций:

* изменять значения цены и остатка определенных товаров в торговых точках;
* добавлять или удалять позиции товаров;
* понятность интерфейса;
* простоту и удобность эксплуатации.

6. Технико-экономические показатели

В данном проекте отсутствуют экономические взаимоотношения. Техническо-экономические показатели не рассчитываются.

7. Стадии и этапы разработки

7.1 Стадии разработки

Разработка производится в соответствии со следующими стадиями:

* оформление технического задания;
* оформление эскизного проекта;
* оформление технического проекта;
* написание программного продукта;
* отладка программного продукта;
* оформление программной документации.

7.2 Этапы разработки

Разработка программного продукта включает в себя данные этапы:

* выбор темы и разработка плана;
* изучение литературы по теме и проведение анализа предметной области;
* проектирование и разработка программного решения;
* написание первой части курсовой работы;
* написание и отладка программы, и написание второй части работы.

8. Порядок контроля и приемки

Проведение тестов на работоспособность. При успешном прохождении тестов на работоспособность передача заказчику. Заказчик проверяет соответствие программы техническому заданию и проверяет содержание пояснительной записки.

Эскизный проект

1. Предварительная разработка структуры входных

и выходных данных

Планируется разработка программного продукта для контроля наличия товаров и их цен. Предварительными входными данными будут:

* индекс выбора действия, при помощи которого пользователь будет выбирать позиции продуктов и действия, связанные с их изменением;
* код товара;
* наименование товара;
* количество оставшегося товара, в котором считается товар как на полках магазина, так и на складе;
* цена товара.

Также среди входных данных будет файл. Он будет содержать позиции товаров построчно. Столбцы товаров будут следующие:

* код товара;
* наименование товара;
* количество товара;
* цена товара.

Для вывода данных планируется построчный вывод всех доступных позиций, заданных в программе. В выводе предварительными данными будут:

* описание текущего состояния программы;
* код товара;
* наименование товара;
* количество оставшегося товара, в котором считается товар как на полках магазина, так и на складе;
* цена товара.

2. Уточнение методов решения задачи

Для решения задачи необходимо будет разработать программу со средством ввода-вывода данных для пользователя и средством хранения данных. Ввод-вывод необходим для предоставления удобного интерфейса пользователю приложения. Будут выводиться состояние программы, подсказки. При выборе состояния вывода будут построчно выводиться позиции из средства хранения данных. В строках будут находиться: код продукта, что будет являться определенной позицией продукта; наименование продукта; количество остатка продукта в торговой точке, причем считаются продукты как на полках, так и на складе магазина; цена товара в данной точке. Также пользователь будет иметь возможность изменять состояние программы путем ввода индекса состояния программы: исходное состояние; состояние просмотра данных; состояние изменения данных; состояние добавления позиций; состояние удаления позиций. В состоянии изменения данных вводится индекс столбца, который необходимо изменить, а затем производится изменение данных. В состоянии добавления данных вводятся номер кода товара, наименование товара, количество оставшегося товара и цену товара. В состоянии удаления позиций пользователь вводит номер кода товара и подтверждает удаление товара для предотвращения случайных удалений.

3. Разработка общего описания алгоритма решения задачи

После запуска программы пользователь видит приветствие, затем выводится экран исходного состояния. В экране исходного состояния пользователю предоставляют изменение состояния программы. При выборе состояния программа выводит необходимый интерфейс для данного состояния. При любом вводе у пользователя будет возможность выйти в исходное состояние с запрашиваемом подтверждением намерений для исключения случаев случайного ввода. При попытке выхода в исходном состоянии с подтверждением намерений для исключения случаев случайного ввода пользователь закроет программу.

При выборе состояния вывода данных программа даст выбор просмотреть все существующие позиции в файле или какую-то либо конкретную. При выборе вывода всех существующих позиций, соответственно, программа выведет все существующие в файле позиции. После этого программа запросит пользователя подтвердить выход в меню. При выборе вывода данных по конкретной позиции программа запросит номер кода товара. При вводе существующего кода программа выведет данные конкретного товара. При вводе несуществующего кода программа выведет текст ошибки о попытке вывода несуществующей позиции. В обоих случаях программа даст пользователю выбор повторить поиск конкретного товара или выйти в меню, то бишь в исходное состояние.

Для состояния изменения данных программа запрашивает код товара. После ввода кода товара программа запрашивает ввод индекса, по которому определяется, какие именно данные нужно изменить нужно изменить. При попытке изменить код товара программа проверит введение данные на существование в файле. Если в файле нет позиции с данным кодом, то программа запросит подтверждение изменения данных. Иначе программа выведет ошибку о существовании данного кода в файле. При попытке изменения количества и цены на отрицательное число программа выведет ошибку о попытке изменения этих данных на отрицательное значение.

Для состояния добавления данных программа запрашивает ввод следующих данных: код товара; наименование товара; количество товара; цена товара. После ввода программа проверяет на существование кода товара в файле. Если такого же кода нет, то позиция добавляется в файл. Если код уже имеется, то программа выводит ошибку о существующей позиции с похожим кодом. В обоих случаях после выполнения действий программа дает выбор добавить еще одну позицию или выйти в меню, то есть в исходное состояние.

В состоянии удаления позиций программа запрашивает код товара. При несуществующем коде программа выведет ошибку и даст выбор повторить операцию удаления позиции или выйти в меню. При существующем коде программа выведет данные данной позиции и запросит подтверждение удаления. После совершения операции программа дает выбор повторить операцию удаления или выйти в меню, то есть в исходное состояние.

Технический проект

1. Уточнение структуры входных и выходных данных

Принцип работы основан на взаимодействии пользователя с консолью программы при помощи ввода инструкций и данных через клавиатуру и вывода данных на монитор.

Среди вводных данных самым главным является файл, расположенный на компьютере. Его путь будет задан только в коде, пользователь не сможет поменять его местоположение через консоль. Временный файл существует только при работе программы. Он нужен для взаимодействия с данными файла, считыванию и записи данных файла. Временный файл будет двумерным динамическим массивом из символьного формата. После ввода данные будут храниться в буфере текстового формата. Для проверки кода товара данные символьного типа из временного файла будут формироваться в текстовый формат и сравниваться с буферным значением. Для проверки на ввод положительного значения для данных о количестве и цене текстовый формат из буфера будет переводиться в целочисленное значение или в значение с плавающей точкой в зависимости от совершаемой операции.

Среди выходных данных имеются текстовые подсказки для пользователя, файл и позиции в файле. Текстовые подсказки являются текстом, написанном в коде. Пользователь никак не сможет каким-либо образом их изменить. Файл на выводе будет формироваться из временного файла, являющийся двумерным динамическим массивом символьного формата. Позиции из файла формируются из временного файла путём сложения в один операнд текстового формата.

2. Разработка алгоритма решения задачи

Программа работает до ввода пользователем индекса «exit» в исходном состоянии. Индекс «exit» в любом другом состоянии за исключением исходного вернет программу в исходное состояние.

Файл, который будет упоминаться дальше, будет читать и записывать при помощи класса ifstream, а функции будут работать с vector с вложенным массивом или vector из класса char. Vector будет временным файлом в программе, который будет хранить все данные из обычного файла для чтения и записи. Именно vector будет упоминаться как «файл».

Программа будет состоять из функций main(), ShowPositions(), RedactPosition(), AddPosition(), DeletePosition(). Названия функций соответствуют названию их состояний.

Функция main() – функция исходного состояния. В ней начинает и заканчивает свою работу программа. В ней будет расположен цикл while, который завершится при вводе пользователем индекса «exit».

Функция ShowPositions() - функция вывода позиций. В ней происходит вывод всех или определенных позиций из файла. Для выбора действия будут использоваться индексы действий «1» и «2» или «ShowAll» и «ShowSpecific», которые будут выполнять действия «показать все позиции» и «показать специфическую позицию по коду товара» соответственно. Нахождение специфической позиции по коду товара выполняется при помощи функции std::find. Если код найден, то выводится позиция. Если код не найден, то выводится ошибка. После выполнения обоих случаев программа опять запросит ввод действий и выдаст подсказку о возможности выйти в исходное состояние. Выйти из данной функции можно в любой момент при помощи кодового индекса «exit».

Функция RedactPositions() – функция редактирования позиций. После начала работы запрашивает код товара. Код товара проверяется на существование в файле при помощи std::find. При не существующем коде выдает ошибку. При существующем коде выводится позиция с данным кодом и далее программа запрашивает индекс позиции, которую нужно изменить. На ввод принимают числа «1», «2», «3» и «4» или «ChangeCode», «ChangeName», «ChangeStock» и «ChangePrice», которые выбирают изменение кода товара, наименование товара, количество товара и цену соответственно. При попытке изменить код на существующий, ввод отрицательного значения для количества товара и цены программа выдаст ошибку. После ввода изменения программа запрашивает подтверждение изменений. После выполнения программа даёт выбор повторить операцию изменения или выйти в меню, которые срабатывают на ввод «Yes» и «No» соответственно. Также выйти из данной функции можно в любой момент при помощи кодового индекса «exit».

Функция AddPosition() – функция добавления позиций. После начала работы сразу запрашивает ввести код товара, который надо добавить. Ищет введенный код на наличие в файле при помощи std::find. Если введенный код уже существует в файле, то программа выдаст ошибку и даст возможность ввести код еще раз. Иначе дальше программа запросит ввод наименования, количества товара и цены товара. Перед добавлением запрашивает подтверждение правильности введенных данных. После выполнения операции дает выбор ввести еще одну позицию или выйти в меню при вводе индексов «Yes» и «No» соответственно. Также выйти из данной функции можно в любой момент при помощи кодового индекса «exit».

Функция DeletePositions() – функция удаления позиций. После начала работы сразу запрашивает код товара. Если при помощи std::find не будет найдена позиция с соответствующим кодом в файле, то программа выдаст ошибку. Иначе, программа выведет все данные данной позиции из файла и запросит подтверждение удаления. Здесь программа будет ждать ввод на подтверждение и отмену или «Yes» и «No» соответственно. После выполнения операции программа дает возможность выбрать удаление еще одной позиции или выход в меню или «Yes» и «No» соответственно. Также выйти из данной функции можно в любой момент при помощи кодового индекса «exit».

3. Определение формы представления входных и выходных данных

Изначально перечислим структуру входных данных: файл с данными, который является классом ifstream; временный файл для считывания перед операцией, хранения и изменения данных во время операции и записи после операции, который будет состоять из вектора с вложенным вектором класса char; ввод индекса состояния и действия класса string.

Структура вывода: файл с данными класса ifstream, записываемый в формате txt; вывод позиции в string.

В самом файле данные будут находиться через пробел.

4. Определение семантики и синтаксиса языка

Разработка программы планируется на последней версии языка C++ по состоянию на 22 ноября 2021 года в IDE Microsoft Visual Studio.

5. Разработка структуры программы

Для разработки данной программы будет применяться метод нисходящей разработки с конструктивным подходом при необходимости.

Главной функцией является main(). Начало и конец работы связаны именно с этой функцией, как и переходы в другие функции. Программа будет заканчивать свою работу, как только пользователь разорвёт цикл while вводом в консоль «exit».

Функция ShowPositions(), в которую программа переходит из главной функции при вводе в консоль «Show positions», будет показывать все позиции или определённую позицию в файле путём образования одного текстового операнда из символьных значений одной строки. Полностью алгоритм работы функции описан в разделе 2 «Разработка алгоритма решения задачи» данной главы. Переход в главную функцию совершается при вводе «exit» в любое время или при вводе «no» после завершения всех операций в функции и запроса выбора следующих действий от пользователя.

Функция RedactPositions(), в которую программа переходит из главной функции при вводе в консоль «Redact positions», будет изменять определенную позицию путем переписи определенных данных выбранной позиции. Полностью алгоритм работы функции описан в разделе 2 «Разработка алгоритма решения задачи» данной главы. Переход в главную функцию совершается при вводе «exit» в любое время или при вводе «no» после завершения всех операций в функции и запроса выбора следующих действий от пользователя.

Функция AddPosition(), в которую программа переходит из главной функции при вводе в консоль «Add positions», будет добавлять позиции товаров в файл. Полностью алгоритм работы функции описан в разделе 2 «Разработка алгоритма решения задачи» данной главы. Переход в главную функцию совершается при вводе «exit» в любое время или при вводе «no» после завершения всех операций в функции и запроса выбора следующих действий от пользователя.

Функция DeletePositions(), в которую программа переходит из главной функции при вводе в консоль «Delete positions», будет удалять выбранные позиции в файле. Полностью алгоритм работы функции описан в разделе 2 «Разработка алгоритма решения задачи» данной главы. Переход в главную функцию совершается при вводе «exit» в любое время или при вводе «no» после завершения всех операций в функции и запроса выбора следующих действий от пользователя.

6. Окончательное определение конфигурации технических средств

Для разработки будет использоваться ноутбук с процессором   
Intel 7-7700HQ и 16 гигабайтами оперативной памяти с установленным IDE Microsoft Visual Studio. Для работоспособности программы не будет требоваться значительная вычислительная мощность. В качестве проверки компьютер должен будет способен работать на операционной системе Windows 10 без особых проблем, осложняющих работу пользователя с программой и наличием оборудования для ввода инструкций с данными и вывода данных. Самые важные критерии: операционная система компьютера должна работать стабильно и не выдавать критических ошибок; компьютер не должен зависать, то есть не должен быть сильно нагружен и брать много времени на выполнение операций.