**Министерство науки и высшего образования Российской Федерации**

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**Московский приборостроительный техникум**

Выпускная квалификационная работа

(Дипломная работа)

На тему: Разработка информационной системы выполнения анализа и устранения аномальных ситуаций с прайсом товара

ИВАНОВА АНДРЕЯ ДМИТРИЕВИЧА

Студент (-ка) 4 курса группы И-1-16

по специальности 09.02.05 Прикладная информатика (по отраслям)

для присвоения квалификации: техник – программист

Форма обучения:  очная

Руководитель: / /

(подпись)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Консультант: / /

(при наличии) (подпись)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Студент (-ка): / /

(подпись)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Допущена к защите

Распоряжение от «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. № \_\_\_\_\_\_\_\_\_\_\_\_\_

2020

Оглавление

[ВВЕДЕНИЕ 4](#_Toc41421492)

[ОБЩАЯ ЧАСТЬ 5](#_Toc41421493)

[1.1 Цель разработки 5](#_Toc41421494)

[1.2 Средства разработки 5](#_Toc41421495)

[СПЕЦИАЛЬНАЯ ЧАСТЬ 6](#_Toc41421496)

[2.1 Постановка задачи 6](#_Toc41421497)

[2.2 Внешняя спецификация 6](#_Toc41421498)

[2.3 Проектирование 6](#_Toc41421499)

[2.4 Результат работы программы 6](#_Toc41421500)

[ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 7](#_Toc41421501)

[3.1 Инструментальные средства разработки 7](#_Toc41421502)

[3.2 Отладка программы 7](#_Toc41421503)

[3.3 Защитное программирование 7](#_Toc41421504)

[3.4 Характеристики программы 7](#_Toc41421505)

[ЗАКЛЮЧЕНИЕ 8](#_Toc41421506)

[Список использованных источников и литературы 9](#_Toc41421507)

Приложение А. Текст программы

Приложение Б. Руководство пользователя

# ВВЕДЕНИЕ

Во введении оценивается современное состояние и уровень автоматизации предметной области, разъясняется значение автоматизации рассматриваемого процесса для соответствующего производства (отрасли промышленности и т.п.), обосновывается актуальность темы, формируется цель и задачи дипломной работы.

Введение является частью пояснительной записки, где описывается проблематика предметной области, современный уровень развития отрасли, к которой принадлежит дипломная работа.

# ОБЩАЯ ЧАСТЬ

## Цель разработки

Следует указать, с какой целью делалась разработка.

Например, одна из приведённых ниже:

* автоматизация процесса…;
* повышение точности расчетов…;
* повышение качества обслуживания торговых точек;
* упрощение формирования отчётов по популярным аномальным ситуациям с прайсом товара;
* обеспечение…;
* организации….;
* совершенствование…;
* сокращение времени на выполнение формирования дежурных инженерных групп;
* сокращение персонала при выполнение каких-либо работ…;
* освобождение персонала от вредной, утомительной, рутинной работ подсчёта популярных проблем;

## Средства разработки

Следует кратко указать необходимую для разработки программы вычислительную систему:

* технические средства (ее компоненты и их характеристик),

программные средства (операционная система, инструментальные средства разработки) и аргументировать их выбор.

# СПЕЦИАЛЬНАЯ ЧАСТЬ

## Постановка задачи

Постановка задачи предусматривает ответы на следующие вопросы:

1. Описание задачи

Здесь уточняется постановка задачи, формулируются условия задачи в общем виде и определяются все функции программы, приводятся описания этих функций.

Если разрабатывается игровая программа, то описывается сценарий игры.

Если разрабатывается программа по обработке данных, то описываются функции по обработке данных и их особенности.

Если в приложении предполагается использование разграничение доступа, то необходимо указать количество уровней доступа и изложить что кому доступно.

Если система многопользовательская, то необходимо здесь указать.

Для наглядности здесь уместно использовать диаграммы унифицированного языка моделирования UML (Unified Modeling Language), например, диаграммы вариантов использования, диаграммы деятельности, диаграммы классов. Последние две диаграммы в зависимости от детализации могут использоваться на разных этапах проектирования.\*

1. Входные и выходные данные

Для входных, выходных данных указываются:

* имя - физический смысл,
* тип (физический смысл типа: целое число, вещественное число, строка символов и т.д., а не обозначение типа на языке программирования),
* размер в байтах,
* диапазон изменения значений,
* точность представления (в случае математических вычислений),
* [структура/формат], [объем],
* форма ввода/вывода данных (бланк, таблица, график и т.д.),

По физическому смыслу типа и размеру в байтах в дальнейшем определяется тип данных на уровне языка программирования.

* данные, которые надо получить,
* требования к работе (развернутые требования к программе: логический контроль информации, разработка пользовательского интерфейса, и т.д.).

1. Модель оценки качества

Данный пункт описывает качество исполнения программного продукта в соответствии с выдвинутыми требованиями к разрабатываемому программному продукту ВКР. Критерии оценивания оформляются следующим образом, представленным в таблице 1.1., в соответствии с международным стандартом оценивания качества ISO 9126.

Внимание**!**

Вся общая сумма критерием не должна превышать 100 баллов. Если рад критериев не может быть оценен из-за специфика программного продукта, необходимо ниже представленной таблицы представить обоснование невозможности описания того или иного атрибута или метрики.

Таблица 1.1. Модель оценки качества

| Название атрибута | Метрика | Описание метрики | Оценка | Описание |
| --- | --- | --- | --- | --- |
| Название атрибута 1 | Метрика 1 | Описание метрики 1 | 5 | Описание оценки 1 |
| 4 | Описание оценки 2 |
| 3 | Описание оценки 3 |
| 2 | Описание оценки 4 |
| 1 | Описание оценки 5 |
| 0 | Описание оценки 6 |
| Метрика 2 | Описание метрики 2 | 5 | Описание оценки 7 |
| 4 | Описание оценки 8 |
| 3 | Описание оценки 9 |
| 2 | Описание оценки 10 |
| 1 | Описание оценки 11 |
| 0 | Описание оценки 12 |
| … | … | … | … | … |

## Внешняя спецификация

Разработка внешней спецификации - это процесс описания планируемого поведения программы с точки зрения потенциального пользователя. Целью этого процесса является конкретизация внешних взаимодействий будущей программы без детализации внутреннего устройства.

Внешняя спецификация состоит из:

* метод, (метод тестирование)
* тесты, ( тест-план, тест – кейсы)

1. Метод

Здесь необходимо указать название метода, используемого математического аппарата (защиты данных, защитного программирования, базы данных, технологии программирования).

Для математического аппарата если связь между выходными данными и исходными выражена аналитическими зависимостями, формулами, то необходимо привести математические выкладки. Если используется один из численных методов, то необходимо привести краткое описание метода, его математический аппарат.

Метод решения задачи приводится на логическом уровне.

Для базы данных необходимо сказать о нормализации данных, указать методы обработки (например: индексная сортировка, логическое удаление, восстановление данных и т.д.)

Если используется защита данных или программы, то указывается метод защиты.

Если при разработки принимаются во внимание технологии, принципы, методы, методологии разработки, то здесь необходимо указать.

1. Тесты

Данный пункт состоит из 3 частей:

- Перечень функций программного обеспечения – в данной части описывается весь функционал программного продукта. Стандартные функции БД (Добавление, изменение, удаление), авторизация, шифрование, аутентификация, подключение к источнику данных, функции взаимодействия с операционной системой и т.п.

- Тест план – здесь описывается план проведения тестирования программного продукта, он представлен в виде таблице ниже. В тест плане должны быть затронуты все области программного продукта перечисленные в пункте выше (перечень функций ПО).

Таблица 1.2. Тест план

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Область функции | Приоритет | Стратегия тестирования | Количество повторов | Риски | Статус |
| Функции БД | 3 | Полная проверка добавления данных всех полей | 50 | Часть функций не работает корректно | Проектирование |

«Область функций» описывается область тестирования;

«Стратегия тестирования» кратко описывает как будет тестироваться программный продукт;

В «количестве повторов» указывается примерное количество тестов в данной области тестирования;

«Риски» описывают возможные риски данной области.

- Тест кейсы – могут быть представлены в виде таблицы c тест кейсами или описанием unit-тестов. В обоих случаях в здесь тестируются все функции программного продукта описанные в пункте «Перечень функций программного обеспечения».

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер теста | Вид теста | Действие | Ожидаемый результат | Результат теста |
| 1 | позитивный | Операционная система Windows 10  Раскладка клавиатуры  RU  Запуск приложения | Должно открыться приложение  Окно по центру  В диспетчере задач должен отображаться  Иконка приложения (смысловая)  Название окна на русском языке «Информационная система» | … |

«Вид теста» - описывает вид проводимого теста либо «позитивный», либо «негативный»;

«Действие» - описывает порядок действий пользователя, а так же внешние факторы влияющие на тест в контексте тестируемой функции;

«Ожидаемый результат» - описывает реакция программы на каждое действие пользователя;

«Результат теста» – описывается результат тестирования, заполняется только если тест провален, если тест пройден успешно, остается пустым.

Внимание!

На основе приведенного набора тестов будет строиться материал подраздела "Результаты работы программы".

## Проектирование

Цель проектирования (внутреннего проектирования) - создать ясную и относительно простую внутреннюю структуру программы, данных, пользовательского интерфейса.

При разработке проекта могут использоваться любые технологии и их комбинации:

* структурное программирование,
* объектно-ориентированное программирование,
* визуальное программирование.

В данном разделе должны быть определены не только те технологии, которые определяют структуру приложения, но и те которые не отражаются структуру программы (платформенно-независимые, многокомпонентные и др.), но используются в проекте, например:

* другие приложения,
* другие ресурсы,
* серверы: внутренний, локальный или удаленный.

Все вышесказанное должно быть отражено в виде схем, рисунков, поясняющих процессы, а также должно быть упомянуто в разделе «Внешняя спецификация» («Метод»).

На этапе проектирования создаются схемы, в которых в отдельные части будут органично вписываться, и посредством которых, можно управлять написанием этих отдельных частей программы.

Проектирование можно разделить на укрупненное и детальное. Проектирование производить соответствующими известными технологиями и методами. Результатом детального проектирования являются алгоритмы отдельных блоков программы.

Проектные схемы есть конечный продукт процесса проектирования. В раздел входит:

* схема архитектуры программы,
* схемы данных,
* функциональная схема задачи,
* структурная схема программы,
* схема пользовательского интерфейса,
* укрупненный алгоритм,
* алгоритмы (отдельных блоков программы) и т.д.

1. Схема архитектуры программы

В случае реализации файл-серверных, клиент-серверных, в том числе трёхуровневой системы, указываются выбранный вид архитектуры, представляется схема архитектуры программы (например, в виде диаграммы развёртывания UML). В схеме указывается серверная и клиентская части приложения их свойства и функции.

В случае многослойных систем (когда система разбивается на уровни бизнес-логики, нижестоящий уровень инфраструктурный, например связь с СУБД, вышестоящий прикладной, связывающий бизнес-логики с пользовательским интерфейсом), необходимо представить схему архитектуры с уточнением свойств и функций.

1. Схемы данных

Здесь необходимо указать все информационные блоки программы и пояснить формат каждого блока. Возможные варианты:

a) в случае сложного формата данных представить схему структуры (формата) данных, пояснив схему;

б) схематично отразить путь данных при решении задачи и определить этапы обработки;

в) логическая (инфологическая) и физическая (даталогическая) схемы данных,

в) схемы логической взаимосвязи данных типа сущность-связь (рекомендуется при проектировании баз данных),

г) диаграммы обработки и т.д.

Схемы данных выполняются в соответствии с ГОСТ 19.701-90.

1. Функциональная схема задачи

Здесь надо отразить декомпозицию задачи (составные части решаемой задачи), сохраняя ее физический смысл. Представление зависит от предметной области задачи и используемого вида программирования (структурного или объектно-ориентированного).

При структурном программировании схема представляет иерархическую структуру разбиения задачи на подзадачи.

При объектно-ориентированном программировании составляется схема иерархии объектов (диаграмма объектов UML на логическом уровне) и пояснения к схеме в виде таблицы. В таблице приводятся данные каждого объекта:

* имя - физический смысл (соответствующий на схеме),
* свойства объекта ( индивидуальные особенности),
* действия (методы) объекта (характеризующие его индивидуальные особенности).

Рекомендуется приводить в схемах не менее трех уровней детализации.

1. Структурная схема программы

Описать принцип построения программы: наличие модулей, процедур и функций, их назначение и взаимосвязь. Информацию можно представить в виде структурной схемы и таблицы.

В случае применения объектно-ориентированного программирования представляется на программном уровне иерархическая схема объектов (диаграмма объектов UML на физическом уровне)и таблица, поясняющая схему. Четко разграничить объекты на используемые, доработанные (или "перекрытые") и собственные (разработанные автором программы). Привести также таблицу полей и методов для объектов, выполненных автором.

Схема может быть укрупненной, в которой указывается иерархия объектов, и детальной, где указываются свойства (характеристики) и действия объектов. Рекомендуется представлять в схемах не менее трех уровней детализации.

Если объектов много, то имеет смысл сначала указать укрупнённую схему объектов, а затем детальную схему объектов.

1. Схема пользовательского интерфейса

В схеме представить навигацию по программе (маршрут работы программы) (все имеющиеся элементы пользовательского интерфейса и последовательность их появление). Если «окна» пользовательского интерфейса многофункциональные, то необходимо представить структуру этих окон.

1. Укрупнённый алгоритм

Представление алгоритма выполняется в виде схемы или словесного описания по согласованию с руководителем ВКР.

1. Блок-схема

Представление блок-схемы алгоритма выполняется традиционным способом в соответствии с ГОСТ 19.701-90. Можно представить блок-схему алгоритма как одного из блоков основного модуля, так отдельного блока программы из структурной схемы п.2.3.2. (по согласованию с руководителем проекта).

Перед блок-схемой необходимо описать появляющиеся в дальнейшем переменные, используемые процедуры/функции, назначение которых сразу не ясно.

Схема взаимодействия объектов (какой объект использует другие объекты) при объектно-ориентированном программировании.

3.4.4. Результаты работы программы

Результаты работы программы должны соответствовать тестам «Внешней спецификации» подразделов «Тесты» и «Контроль целостности данных».

Работа программы по этим тестам должна быть проиллюстрирована, должен быть сделан вывод о работоспособности программы и выполнении всех функций приложения, согласно п.п.2.2.

Если материалов много (более 2-3 страниц), то необходимо сделать ссылку на приложение Б «Руководство пользователя», где более подробно представлены результаты работы программы.

## Результат работы программы

# ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

## Инструментальные средства разработки

Перечислить использованные инструментальные средства и их компоненты. (Развить тему в случае применения средств, не изучаемых в техникуме.)

## Отладка программы

В этой части выполняется подробное описание процесса отладки с указанием средств отладки (методических и инструментальных) и способов отладки.

Здесь должна быть представлена статистика ошибок и приведены примеры наиболее распространённых ("любимых") ошибок.

## Защитное программирование

Защитное программирование включает защиту программы и защиту от ошибок.

*Защита программы*

Защитное программирование - это методология защиты от несанкционированного доступа к программе и/или данным, а также идентификация программы с целью защиты авторства.

Развить тему в случае необходимости применения. В этом случае указать причину применения и решение проблемы.

*Защита от ошибок.*

Здесь необходимо указать, где применялась защита от ошибок, для подтверждения привести фрагменты программы используемой методологии и/или сделать ссылку на программу.

Например: имя модуля – имя процедуры/функции - номер строки (в этом случае необходимо, чтобы строки программы были пронумерованы).

## Характеристики программы

Характеристики программы с точки зрения программиста включают в себя:

* число модулей (внешних), библиотек (своих);
* суммарный размер исходного текста в строчках;
* используемые модули библиотеки системы программирования (как стандартные модули, так и расширенные модули);
* используемые заимствованные модули, библиотеки;
* привести таблицу наименования, назначения и размера каждого (своего) модуля;
* указать дополнительные файлы, необходимые для сборки программы;

указать переносимость программы на другие платформы и т.д.

# ЗАКЛЮЧЕНИЕ

В заключении ПЗ кратко описываются те решения, которые были приняты в разработке, их техническая целесообразность. А также предложения по внедрению в производство.

В заключении подводятся основные итоги: указывается, какими новыми знаниями овладели в процессе работы над проектом, делаются выводы о работоспособности программы на основании результатов отладки и тестирования программы, об актуальности разработки и возможности дальнейшего её применения и развития.

Например:

В изложенной дипломной работе была изучена актуальная проблема

автоматизации ....

решения ....

обеспечения ....

организации ....

записи ....

Была разработана программа, цель использования которой ....

# Список использованных источников и литературы

Обязательный для ВКР раздел. Здесь необходимо указать все источники, с помощью которых разработана дипломная работа, т.е. те на которые в тексте "Пояснительной записки" автор работы делает ссылки.

В списке использованных материалов приводятся полные библиографические сведения о книгах, статьях и других источниках. Список использованных источников и литературы составляется в алфавитном порядке или в порядке их упоминания в тексте. Примеры правильного оформления источников - смотри список рекомендованных источников и литературы.