**СОДЕРЖАНИЕ**

[СОДЕРЖАНИЕ 4](#_Toc451879757)

[ВВЕДЕНИЕ 5](#_Toc451879758)

[1 АНАЛИЗ И ПОСТАНОВКА ЗАДАЧИ 6](#_Toc451879759)

[1.1 Описание существующих аналогов 6](#_Toc451879760)

[1.2 Формирование рекомендаций по созданию АСОИ 13](#_Toc451879761)

[1.3 Постановка задачи на создание АСОИ 13](#_Toc451879762)

[2 НАЗНАЧЕНИЕ, СОСТАВ И СТРУКТУРА СИСТЕМЫ 15](#_Toc451879763)

[2.1 Назначение АСОИ 15](#_Toc451879764)

[2.2 Проектирование программного обеспечения АСОИ 16](#_Toc451879765)

[2.2.1 Требования к программному обеспечению 16](#_Toc451879766)

[2.2.2 Инструментальные средства для реализации программного обеспечения 19](#_Toc451879767)

[2.2.3 Проектирование интерфейса пользователя с системой 21](#_Toc451879768)

[3 РАЗРАБОТКА АСОИ 23](#_Toc451879769)

[3.1 Создание Activity 23](#_Toc451879770)

[3.2 Создание элементов Activity 29](#_Toc451879771)

[3.3 Создание переходов между Activity 31](#_Toc451879772)

[3.4 Создание галереи изображений 33](#_Toc451879773)

[4 РЕАЛИЗАЦИЯ И ИСПЫТАНИЕ АСОИ 36](#_Toc451879774)

[4.1 Загрузка и установка приложения 36](#_Toc451879775)

[4.2 Реализация и работоспособность Splash Screen 38](#_Toc451879776)

[4.2 Реализация и работоспособность Navigation Drawer 40](#_Toc451879777)

[4.5 Реализация и работоспособность кнопок 44](#_Toc451879778)

[4.6 Реализация и работоспособность просмотра коллекций 45](#_Toc451879779)

[5 РАСЧЕТ ЭКОНОМИЧЕСКИХ ПОКАЗАТЕЛЕЙ 48](#_Toc451879780)

[5.1 Расчет объема функций программного обеспечения 48](#_Toc451879781)

[5.2 Расчет заработной платы разработчиков программного обеспечения 50](#_Toc451879782)

[5.3 Расчет себестоимости программного обеспечения 51](#_Toc451879783)

[5.4 Расчет отпускной цены программного обеспечения 56](#_Toc451879784)

[6 ОХРАНА ТРУДА И ЭКОЛОГИЧЕСКАЯ БЕЗОПАСНОСТЬ 59](#_Toc451879785)

[7 РЕСУРСО- И ЭНЕРГОСБЕРЕЖЕНИЕ 66](#_Toc451879786)

[ЗАКЛЮЧЕНИЕ 74](#_Toc451879787)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 75](#_Toc451879788)

ВВЕДЕНИЕ

Целью данного дипломного проекта является разработка системы визуального представления продукции в качестве приложения на смартфоны и планшеты с ОС Android. Данная концепция является актуальной, т.к. успех современного делового человека зависит от скорости его работы и возможности реагировать на инновации, повышающие эффективность его деятельности. В связи с широким распространением интернета, мобильные устройства, если они подключены к сети, являются незаменимыми помощниками. Для таких устройств активно разрабатываются мобильные приложения, которые предназначаются не только для развития бизнеса, но и для развлечений, общения, совершения платежей и прочее. Мобильные приложения стали удобным инструментом маркетинга и коммуникации.

Чтобы доказать успешность этого технического новшества, достаточно назвать несколько преимуществ, даже особо не вдаваясь в их привлекательную и своеобразную специфику:

* удобство – мобильные приложения очень быстро загружаются, избавляя от ожидания загрузки;
* возможность работы в режиме офлайн – очень удобно в случаях аварийного отсутствия сети, как только связь восстановится, все изменения автоматически загружаются в приложение. Кроме того, это экономит деньги;
* большинство продуктов создаются для определенных моделей телефонов – мобильные приложения под Android предоставляют пользователю максимально комфортное управление с учетом специфики аппарата;
* конкуренция в сфере мобильного маркетинга не слишком высока, преимущество получают компании, предложившие свои услуги первыми;
* мобильные приложения позволяют не только вовлечь пользователей в интерактивные процессы, но и могут стать рекламными носителями.

Мобильное приложение – это уникальное средство, которое пока обделено вниманием. Однако его эффективность доказывает, что отношение к нему вскоре кардинальным образом изменится.

1 АНАЛИЗ И ПОСТАНОВКА ЗАДАЧИ

1.1 Описание существующих аналогов

Представление продукции ЗАО «Инволюкс» в настоящее время существует в бумажном и электронном (сайт) виде. Рассмотрим оба варианта:

Бумажная версия – представлена в виде каталога каждой коллекции в отдельности, т.е. одна коллекция – один каталог. Значительный недостаток бумажной версии в том, что она занимает физическое место и не всегда находится «под рукой», нет конкретной классификации продукции, а это немаловажно в поиске нужной информации.

Электронная версия – представлена в виде сайта. Появление сайта значительно упростило поиск нужной информации, пропала необходимость хранить каталоги коллекций, которых очень много. Недостаток в том, что у сайта нет мобильной версии и при просмотре сайта приходится увеличивать и уменьшать представленное браузером содержимое, поэтому для корректной работы с каталогом нужен персональный компьютер с доступом в интернет.

И так, для максимального удобства для потенциальных клиентов требуется следующее:

* минимальная затрата для выбора продукции, т.е. выбор продукции не должен зависеть от каких либо ресурсов, в частности интернет;
* удобство выбора - отсутствие лишних действий при просмотре и выборе продукции;
* компактность – свойство, позволяющее без труда получить доступ к интересующей информации.

На основе исследования, изучив недостатки и преимущества, вывод – для улучшения представления продукции ЗАО «Инволюкс» требуется электронный каталог в виде мобильного приложения. Мобильное приложение содержит все нужные качества: не требует постоянного подключения к интернету (лишь разово для скачивания приложения на устройство), отсутствие лишних действий при просмотре, а также компактность, что позволяет получить доступ к интересующей клиента информации в любое место и время.

Для реализации мобильного приложения, следует обратиться к статистике и выбрать платформу, поддерживающую данное приложение. Статистика показывает, что для русскоязычных стран (СНГ), с которыми ведёт свою коммерческую деятельность предприятие, присущи девайсы на платформе Android, т.к. доля пользователей использующих устройства на Android составляет 70%.

Плюсом Android является открытость этой операционной системы. Разнообразные программы и игры на Android пишут и сторонние разработчики, поэтому их существует очень и очень много. В повседневной жизни интерфейс операционной системы оказался очень удобен. Имеет значение и то, что работают приложения для Android очень быстро.

Самовольные перезагрузки устройств и неожиданные зависания встречаются крайне редко и могут быть устранены перепрошивкой устройства или ремонтом специалистами. Можно без каких-либо последствий устранить негативные последствия случайных движений пальцев по экрану, просто отменив их.

Особенным вниманием Android пользуется у людей, стремящихся к качеству мультимедиа, видео и фотоматериалом. Огромное количество форматов, которое поддерживается устройствами на базе этой операционной системы, также является огромным плюсом. Практически все приложения работают в тесной связке с глобальной сетью, поэтому легко синхронизируются друг с другом и обновляются. Сами устройства подстраиваются под владельца, поэтому управлять ими очень легко.

Для сравнения выберем каталог известного производителя мебели IKEA (см. рисунок 1.1).



Рисунок 1.1 – Информация из Play Market

Как видно из информации на странице приложения в Play Market, каталог IKEA имеет 3.8 звёзд из 5. Ознакомимся с отзывами о приложении (см. рисунок 1.2).

Как видно из отзывов, приложение работает не стабильно и возможно не устраивает своей функциональностью. Загрузив и открыв приложение на смартфоне, можно заметить, что приложение имеет Splash Screen (форма-заставка). Основная задача формы-заставки – это отображение какого-либо изображения, в то время, пока загружается главная форма приложения. Также на форме-заставке могут отображаться сведения о состоянии загрузки данных, модулей, которые использует приложение.

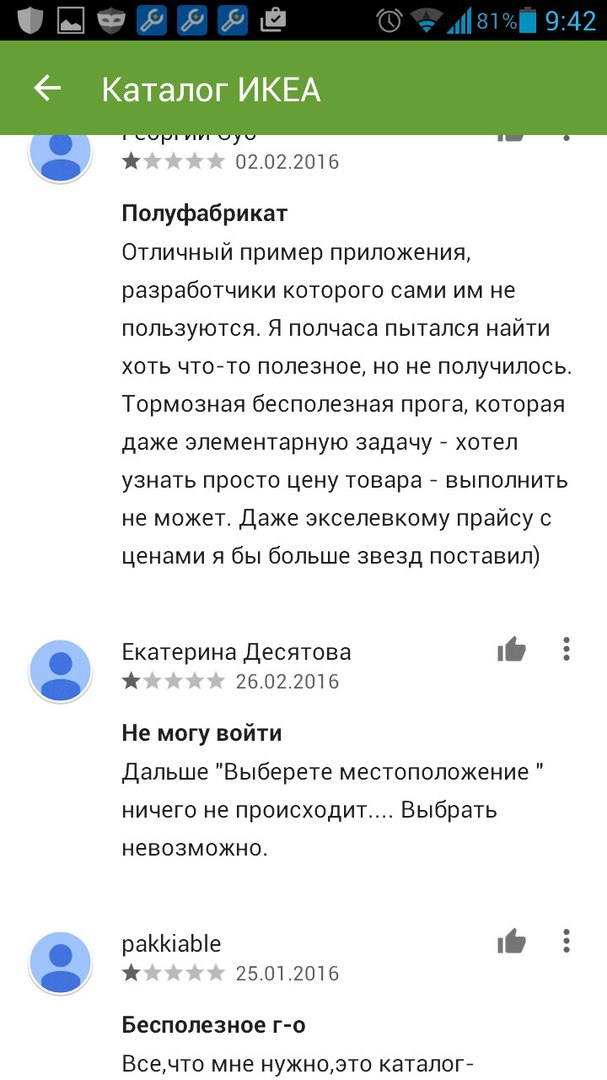
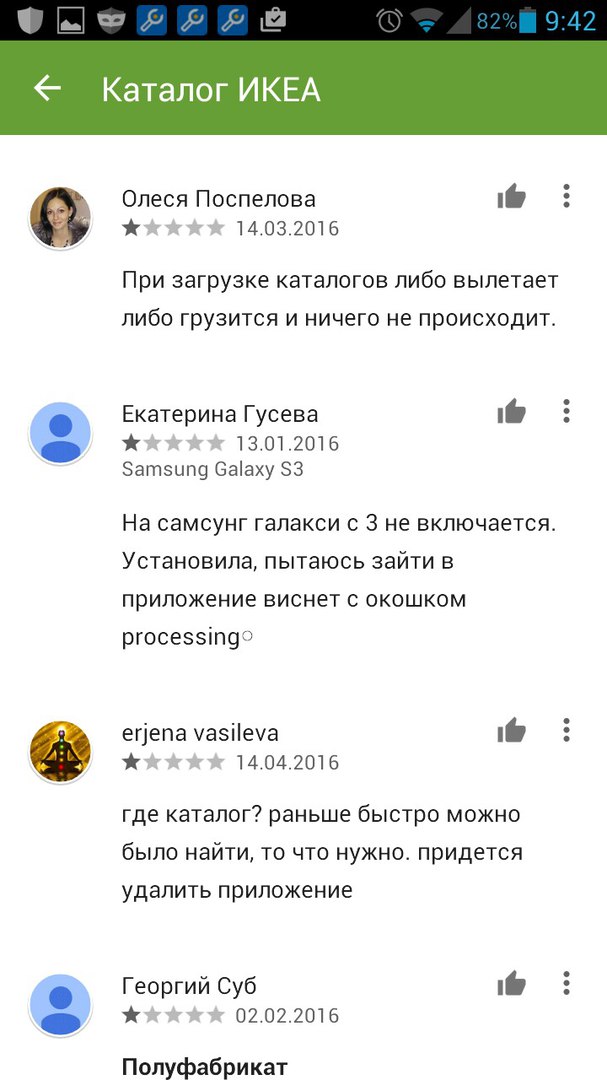
 

Рисунок 1.2 – Отзывы о приложении «Каталог IKEA»

После того, как приложение будет загружено, форма-заставка скрывается, а главная форма выходит на первый план. Есть несколько видов форм-заставок. В основном, она используется для отображения информации или изображения во время загрузки основного приложения. В другом случае, ее можно использовать для отображения номера версии, информации об авторе, сведений о программе. Следует заметить, что форма-заставка приложения, которое мы рассматриваем, не несёт никакой информации кроме как логотипа компании и компании разработчика (см. рисунок 1.3).



Рисунок 1.3 – Форма-заставка приложения «Каталог IKEA»

После некоторой задержки, загружается форма выбора местоположения и выбора языка. Т.к. для Беларуси у компании IKEA нет ни каталога, ни магазина - выберем Россию. Далее загружается главная форма (см. рисунок 1.4). Сверху приложения рекламный баннер с автоматической прокруткой. Далее следуют каталоги по назначению и годам. Следует заметить, в левом верхнем углу иконку – три горизонтальные полоски, означающие, что существует Navigation Drawer [7].

Navigation Drawer - это панель, которая отображает основные опции навигации приложения по левому краю экрана. Эта панель спрятана большую часть времени, но раскрывается, когда пользователь проводит пальцем от левого края экрана, или на верхнем уровне приложения пользователь прикасается к иконке приложения в панели действий.

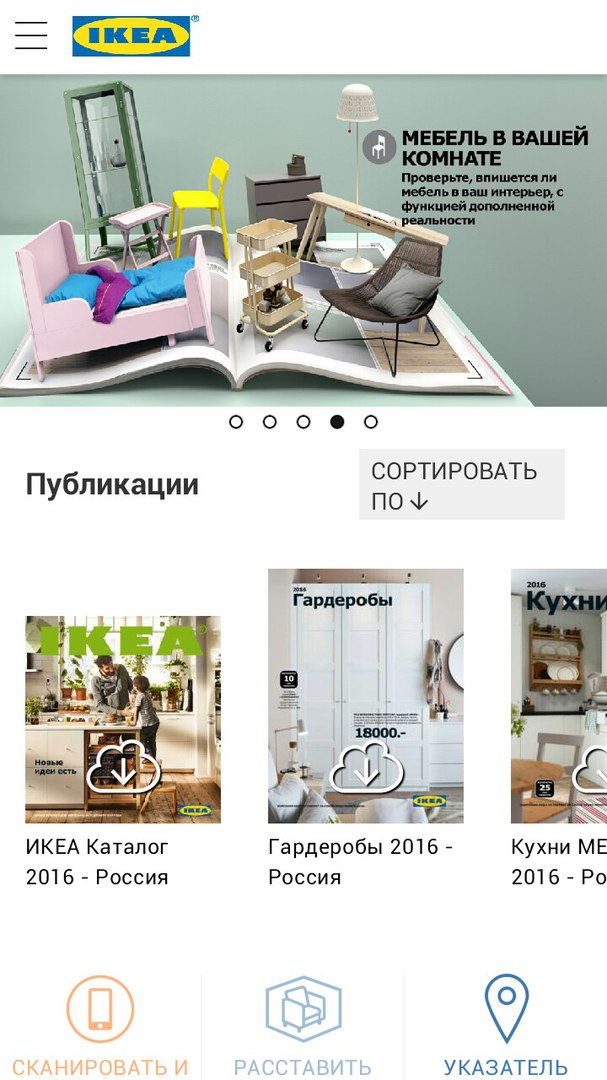


Рисунок 1.4 – Главная форма приложения «Каталог IKEA»

Перейдём по первому же каталогу «ИКЕА Каталог 2016 - Россия». Как видно из изображения на кнопке каталога, информация хранится в облаке и по нажатию будет происходить загрузка ресурсов (см. рисунок 1.5).



Рисунок 1.5 – Загрузка ресурсов

После того как загрузка достигла 100%, выяснилось, что каталог представлен в виде галереи pdf страниц. Задержка при переходе с одной страницы на другую занимает около 2 секунд, что достаточно много, учитывая 2 –х минутную загрузку каталога из интернета в память смартфона. Навигация и поиск в данном приложении весьма сомнительны, т.к. каталог представлен в виде глянцевого журнала, не имеющего какой либо классификации по назначению. То есть получается так, что например информация о детской кроватке может быть расположена и в начале и в середине и в конце каталога, что затрудняет выбор и может запутать покупателя (см. рисунок 1.6).

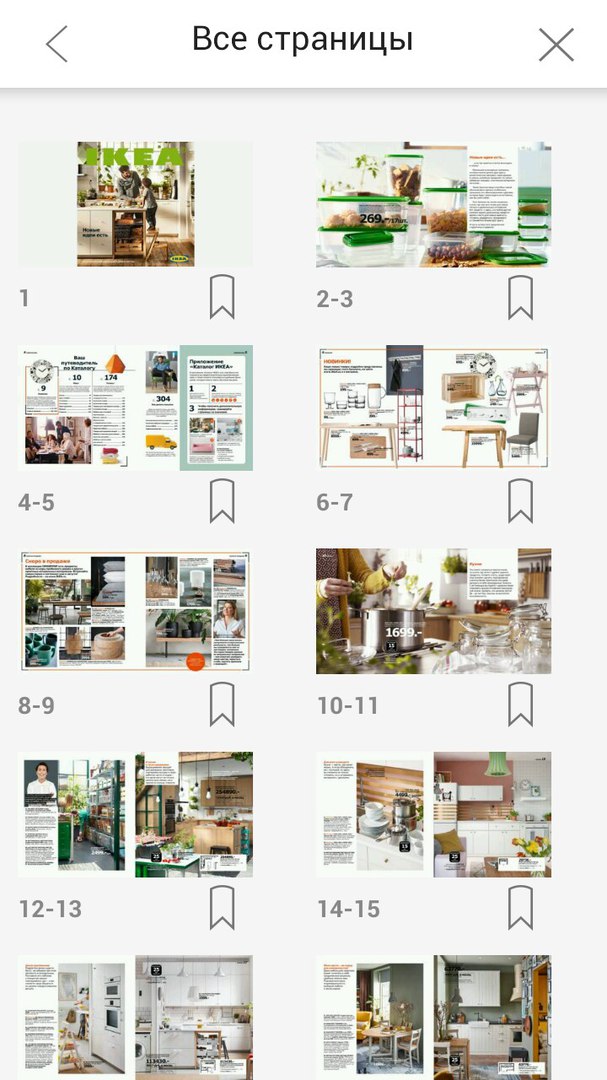


Рисунок 1.6 – Все станицы каталога

На основании рассмотренной информации о аналоге, обозначим основные задачи при проектировании АСОИ:

1. Создание простого и удобного пользовательского интерфейса, без потери в основной функциональности.
2. Минимизировать ожидания пользователей.
3. Доступ пользователей ко всей информации.

1.2 Формирование рекомендаций по созданию АСОИ

Созданное программное обеспечение должно предоставлять пользователю информацию о производимой и продаваемой продукции предприятия. Должно обеспечивать пользователя информацией, где и как приобрести тот или иной продукт. Возможность интуитивно понятно изложить суть работы приложения, а также легко и быстро сориентировать пользователя в структуре приложения. Предоставить графическое и текстовой описание продукции, где графическое описание должно содержать схематическое изображение.

1.3 Постановка задачи на создание АСОИ

Общесистемные требования:

* Система должна иметь иерархическую структуру.
* Приложение должно минимизировано потреблять ресурсы.

**Требования к структуре:**

* Приложение должно быть легко масштабируемо.
* Желательно использование стандартных библиотек и общепринятых паттернов при проектирование архитектуры.

**Требования к техническому обеспечению:**

* Приложение должно работать на устройствах с минимальной тактовой частотой процессора 1 GHz и оперативной памятью не менее 256 Мб.

**Требования к программному обеспечению:**

* Приложение проектируется для работы под управлением ОС Android, но также при необходимости может функционировать на других ОС.
* На устройствах должна быть установлена ОС Android с минимальной версией 4.0.x (Ice Cream Sandwich).

**Требования к лингвистическому обеспечению:**

* В качестве языка программирования рекомендуется использовать Java 8.0.
* Для решения пользовательских задач используется диалоговый пользовательский интерфейс (формы, меню, кнопки и так далее).
* Интерфейс пользователя должен быть интуитивно понятным.

2 НАЗНАЧЕНИЕ, СОСТАВ И СТРУКТУРА СИСТЕМЫ

2.1 Назначение АСОИ

В соответствии с требованиями, которые определены к системе в постановке задачи на дипломное проектирование (пункт 1.3), определим на значение разрабатываемой системы.

Разрабатываемая система предназначена для облегчения навигации в разнообразии продукции ЗАО «Инволюкс».

Потенциальным пользователем данной системы может быть любое физическое или юридическое лицо, у которого есть в наличии девайс с ОС Android. Поскольку по мере ежегодного роста коллекций продукции, все более актуальной становится проблема эффективности поиска и выбора продукции, и все более ощутимым становится падение информативной осведомлённости клиентов.

Разрабатываемая система должна иметь удобную классификацию для поиска информации, визуальное ориентирования в ее потоке. Это позволит увеличить наглядность и интуитивность интерфейса, упростить ориентирование человека в информационном пространстве.

Разрабатываемая система должна предоставить пользователю следующие возможности:

* определить индивидуализацию;
* удобный интерфейс для навигации по назначению продукции;
* возможность просмотра схематического и фотоинформационного представления продукции;
* возможность получить информацию о компании и контактов для приобретения продукции;

Рассмотрим более детально каждую функцию системы.

Определение индивидуализации – это главный процесс в каталоге продукции, т.к. дальнейшее представление продукции зависит от того является ли клиент физическим или юридическим лицом. Например, физическому лицу наиболее вероятно не понадобится информация о мебели для гостиниц или же рецепции для приёмных, так же и наоборот, юридическому лицу – мебель для детской комнаты.

Навигация по назначению продукции так же немаловажна, т.к. это значительно упрощает сориентировать клиента. Например, мебель для спальни, мебель для руководителей, мебель для детской и мебель для персонала – как видно гораздо проще сориентироваться для выбора и просмотра представленной продукции.

Возможность просмотра схематического и фотоинформационного представления продукции позволяет клиенту ознакомится о возможностях размещения продукции в пространстве, цветовая характеристика, компоновка, а также габаритные размеры и способы получения составного вида продукции.

Получение информации о компании и контактов для приобретения продукции, наделяет клиента возможностью легко и быстро получить необходимые сведения.

2.2 Проектирование программного обеспечения АСОИ

2.2.1 Требования к программному обеспечению

Мобильное приложение, для которого нужно реализовать все поставленные задачи, должно быть написано в среде разработки Android Studio.

Android Studio – это интегрированная среда разработки (IDE) для работы с платформой Android, анонсированная 16 мая 2013 года на конференции Google I/O.

Первая стабильная версия 1.0 была выпущена в декабре 2014 года, тогда же прекратилась поддержка плагина Android Development Tools (ADT) для Eclipse.

Особенности Android Studio. Новые функции появляются с каждой новой версией Android Studio. На данный момент доступны следующие функции:

расширенный редактор макетов: WYSIWYG, способность работать с UI компонентами при помощи Drag-and-Drop, функция предпросмотра макета на нескольких конфигурациях экрана;

* сборка приложений, основанная на Gradle;
* различные виды сборок и генерация нескольких .apk файлов;
* рефакторинг кода – процесс изменения внутренней структуры программы, не затрагивающий её внешнего поведения и имеющий целью облегчить понимание её работы;
* статический анализатор кода (Lint), позволяющий находить проблемы производительности, несовместимости версий и другое;
* встроенный ProGuard и утилита для подписки приложений;
* шаблоны основных макетов и компонентов Android;
* поддержка разработки приложений для Android Wear и Android TV;
* встроенная поддержка Google Cloud Platform, которая включает в себя интеграцию с сервисами Google Cloud Messaging и App Engine.

Android - бесплатная операционная система, основанная на Linux с интерфейсом программирования Java.

Android поддерживает 2D и 3D-графику, используя библиотеки OpenGL, а также хранение данных в базе данных SQLite.

Каждое Android-приложение запускается в своем собственном процессе. Поэтому приложение изолировано от других запущенных приложений, и неправильно работающее приложение не может беспрепятственно навредить другим приложениям. Пример разработки приложения представлен на рисунке 1.1.

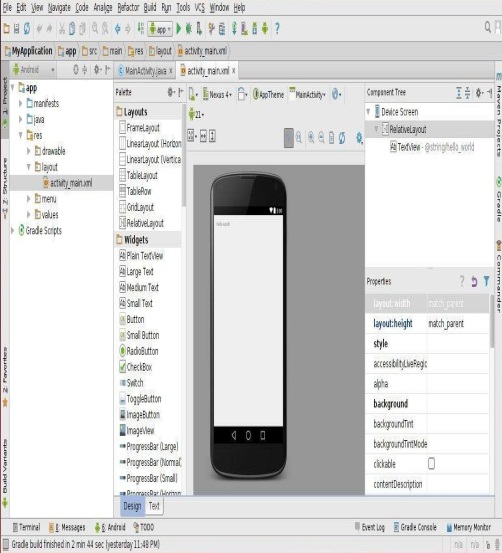


Рисунок 2.1 – Среда разработки Android Studio

Каждое Android-приложение запускается в своем собственном процессе. Поэтому приложение изолировано от других запущенных приложений, и неправильно работающее приложение не может беспрепятственно навредить другим приложениям.

Основные компоненты Android:

Activity (Активность, Деятельность) — представляет собой экран пользовательского интерфейса, который представлен классом Activity и разметкой в виде XML-файла (разметку можно создать и программно, но теперь так не делают). Android-приложение способно состоять из нескольких форм (Activity) и может переключаться между ними во время выполнения приложения.

Intents (Намерения) — асинхронные сообщения, которые позволяют приложению запросить функции из других служб или действий. Приложение может делать прямые запросы службе или действию (явное намерение) или запросить у Android зарегистрированные службы и приложения (неявное намерение). Для примера, приложение может запросить через Intent приложение Контакты для получения данных или запустить браузер при нажатии ссылки в тексте вашего приложения.

Views (Представление, Вид, Элемент управления, Компонент) — пользовательский интерфейс, создаваемый виджетами классов. Это знакомые нам элементы управления: кнопки, текстовые поля, флажки, переключатели и т.п.

Services (Службы) — выполняют фоновые задачи без предоставления пользовательского интерфейса (например, проигрывание музыки). Они могут уведомлять пользователя через систему уведомлений Android.

Content Provider (Контент-провайдер) — предоставляет данные приложениям. С помощью контент-провайдера ваше приложение может обмениваться данными с другими приложениями.

Broadcast Receiver (Приемник широковещательных сообщений/запросов) — принимает системные сообщения и неявные намерения, может использоваться для реагирования на изменение состояния системы. Приложение может регистрироваться, как приемник определенных событий и может быть запущено, если такое событие произойдет.

Для отладки приложений используется эмулятор телефона - виртуальная машина, на которой будет запускаться приложение (см. рисунок 2.2). Эмулятор позволяет отлаживать и тестировать приложения в реальной среде выполнения без необходимости их установки на физическое устройство. Android Virtual Device (AVD) — виртуальное устройство Android. В зависимости от мощности компьютера нужно немного подождать, чтобы сначала загрузился эмулятор. Также можно использовать и реальное устройство или запускать команды командной строки в отдельном окне Terminal.

Окончательную версию приложения желательно проверять на настоящем устройстве. Начиная с Android 4.4, на устройстве (планшет, телефон, часы) нужно активировать режим разработчика. Для этого меню «Настройки», открываем страницу «О телефоне» и щёлкаем семь раз на строчке с номером сборки «Build number». После этого в настройках появится новый пункт «Для разработчиков».

Открыв страницу для разработчиков, нужно включить отладку через USB. Также рекомендуется включать опцию «Не выключать экран». Для некоторых устройств требуется установить отдельный драйвер, для чего стоит искать информацию на сайте у производителей.

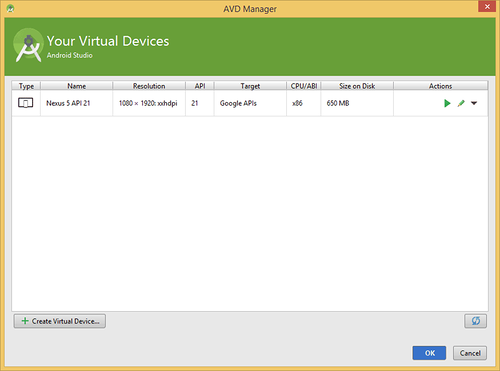


Рисунок 2.2 – Отладка с помощью эмулятора.

### 2.2.2 Инструментальные средства для реализации программного обеспечения

Android — операционная система, основанная на Linux с интерфейсом программирования Java. Это предоставляет нам такие инструменты, как компилятор, дебаггер и эмулятор устройства, а также его (Андроида) собственную виртуальную машину Java (Dalvik Virtual Machine — DVM). Android создан альянсом Open Handset Alliance, возглавляемым компанией Google.

Android использует специальную виртуальную машину, так званую Dalvik Virtual Machine. Dalvik использует свой, особенный байткод. Следовательно, Вы не можете запускать стандартный байткод Java на Android. Android предоставляет инструмент «dx», который позволяет конвертировать файлы Java Class в файлы «dex» (Dalvik Executable). Android-приложения пакуются в файлы .apk (Android Package) программой «aapt» (Android Asset Packaging Tool) Для упрощения разработки Google предоставляет Android Development Tools (ADT) для Eclipse. ADT выполняет автоматическое преобразование из файлов Java Class в файлы dex, и создает apk во время развертывания.

Android поддерживает 2D и 3D графику, используя библиотеки OpenGL, а также хранение данных в базе данных SQLite.

Каждое Android-приложение запускается в своем собственном процессе и под своим собственным userid, который автоматически генерируется Android-ом во время развертывания. Поэтому приложение изолировано от других запущенных приложений, и неправильно работающее приложение не может беспрепятственно навредить другим Android-приложениям.

Android определяет конкретные разрешения для определенных задач. К примеру, если приложение хочет получить доступ в Интернет, оно должно определить в своем файле конфигурации, что оно хотело бы получить соответствующие разрешения. Во время установки Android-приложения пользователю показывается экран, на котором ему нужно дать приложению требуемые разрешения.

Android-приложения описываются файлом «AndroidManifest.xml». В этих файлах должны быть объявлены все активити, службы, приемники и контент-провайдеры приложения. Также он должен содержать требуемые приложением разрешения. Например, если приложению требуется доступ к сети, то это должно быть определено здесь. «AndroidManifest.xml» можно рассматривать, как описание для развертывания Android-приложения.

Атрибут «package» — базовый пакет для следующих элементов Java. Он также обязан быть уникальным, т.к. Android Marketplace только единожды принимает заявку на каждый «package». Следовательно, хорошей привычкой будет использование Вашего обратного доменного имени как «package», для избежания конфликтов с другими разработчиками, «android:versionName» и «android:versionCode» определяют версию Вашего приложения. «versionName» — то, что видит пользователь и может быть любой строкой. «versionCode» должен быть целым, и Android Market использует это для определения, предоставили ли Вы новую версию, чтобы запустить обновления на устройствах, на которых установлено Ваше приложение. Как правило. начинается с 1 и увеличивается на единицу, если Вы выпускаете новую версию приложения.

«Activity» определяет активность, в этом примере указывает на класс «de.vogella.android.temperature.Convert». Для этого класса зарегистрирован фильтр интентов, определяющий, что это активити запускается при запуске приложения (действие android:name=«android.intent.action.MAIN»). Определение категории (категория android:name=«android.intent.category.LAUNCHER» ) определяет, что это приложение добавлено в директорию приложений на Android-устройстве. Значения со знаком «@» ссылаются на файлы ресурсов, которые содержат актуальные значения. Это упрощает работу с разными ресурсами, такими как строки, цвета, иконки, для разных устройств и упрощает перевод приложений.

Часть «uses-sdk» из «AndroidManifest.xml» определяет минимальную версию SDK, на котором можно запускать Ваше приложение. Это предотвращает установку Вашего приложения на устройства с более старой версией SDK.

Каталог «gen» в Android-проекте содержит генерированные значения. «R.java» — генерированный класс, который содержит ссылки на ресурсы из папки «res» проекта. Эти ресурсы содержатся в директории «res» и могут быть значениями, меню, схемами, иконками или рисунками, или анимациями. Например, ресурсом могут быть рисунок или XML-файлы, содержащие определенные строки.

Если Вы создаете новые ресурсы, то соответствующие ссылки будут автоматически созданы в «R.java». Ссылки являются статическими значениями типа int (целочисленными константами), система Android предоставляет методы доступа к соответствующим ресурсам. Например, для доступа к строке с идентификатором ссылки «R.string.yourString» используйте метод getString(R.string.yourString)); Пожалуйста, не пробуйте изменять «R.java» в ручную.

Тогда как каталог „res“ хранит структурированные значения, известные платформе Android, каталог „assets“ может быть использован для хранения любых данных. В Java Вы можете получить доступ к этим данным через AssetsManager и метод getAssets().

### 2.2.3 Проектирование интерфейса пользователя с системой

Пользовательский интерфейс приложения представляет собой иерархическую структуру. После загрузки и установки приложения на устройство, появляется ярлык с легкоузнаваемой иконкой, а также подпись. Иконка и подпись должны с одного взгляда пользователя объяснить суть приложения.

Загрузка приложения сопровождается трёхсекундным Splash Screen - формы-заставки – это отображение какого-либо изображения, а также дополнительной информации о компании и приложении, в то время, пока загружается главная форма приложения. Выбор времени для отображения три секунды, так как этого времени достаточно для, того чтобы пользователь получил всю необходимую информацию. Если время отображения будет меньше, соответственно этого будет недостаточно. Увеличение времени отображение приведёт к тому что пользователь будет долго ожидать загрузки основной формы приложения, что негативно отразится на мыслительном и эмоциональном отношении пользователя к системе.

После загрузки формы-заставки отображается главная форма приложения, которая сразу определяет целенаправленность и индивидуализацию пользователя. То есть если пользователь является физическим лицом, со 100% вероятностью его интересует мебель для дома, никак не для офиса или для гостиниц, и наоборот.

С главной формы приложения, а также из всех наследуемых форм можно открыть Navigation Drawer – боковая панель, которая отображает основные опции навигации приложения по левому краю экрана, либо по правому краю экрана. Боковая панель является весьма удобным способом навигации. Из бокового меню можно перейти:

* на главную форму;
* на форму информации о компании;
* на форму сведений о контактах.

Рассмотрим более детально каждый пункт меню.

Переход на главную форму, всегда удобен, так как иногда надо начать всё сначала, и пользоваться встроенными переходами «назад» - весьма долго и неудобно.

Переход на форму информации о компании, позволяет пользователю в любой момент использования системы, ознакомится о деятельности компании, о её истории, инновациях и развитии.

Переход на форму сведений о контактах, очень важный пункт меню, так как этот пункт позволяет пользователю заказать какое либо изделие или услугу. Также там содержится информация о банковских реквизитах компании, телефон и факс, как головного офиса, так и связаться напрямую с производством. Имеет ссылку на официальный сайт и интернет-магазин.

В главной форме после определения индивидуализации пользователя происходит переход по назначение мебели, то есть, например, для «Мебель для дома» есть возможность выбрать мебель по классификации: для спальни, гостиной, детской и т.д.

# 3 РАЗРАБОТКА АСОИ

Разработка приложения состоит из нескольких этапов:

* cоздание Activity;
* создание элементов Activity;
* создание переходов между Activity;
* создание галереи изображений.

**3.1 Создание Activity**

В среде разработки Android Studio Activity (активность) можно создать двумя способами:

* автоматически;
* вручную.

В данном случае выберем автоматический режим создания Activity (см. рисунок 3.1). Студия предлагает несколько шаблонов активностей. Выбираем Blank Activity – самый простой шаблон.

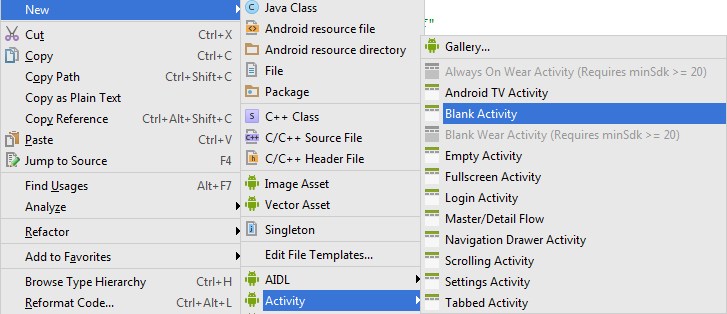


Рисунок 3.1 – Создание активности

Далее студия открывает окно в котором следует заполнить несколько полей ввода (см. рисунок 3.2):

* название активности;
* название файлов разметки;
* иерархия.

По умолчанию изменение названия активности, автоматически отражается на названии файлов разметки. Если это не устаивает можно вручную отредактировать название файлов разметки.

Поле ввода Hierarchical Parent по умолчание является пустым. Если создаваемая активность в иерархическом представлении не является родителем поле оставляется пустым [1]. Если же требуется указать родителя, студия предложит все активности проекта для выбора.

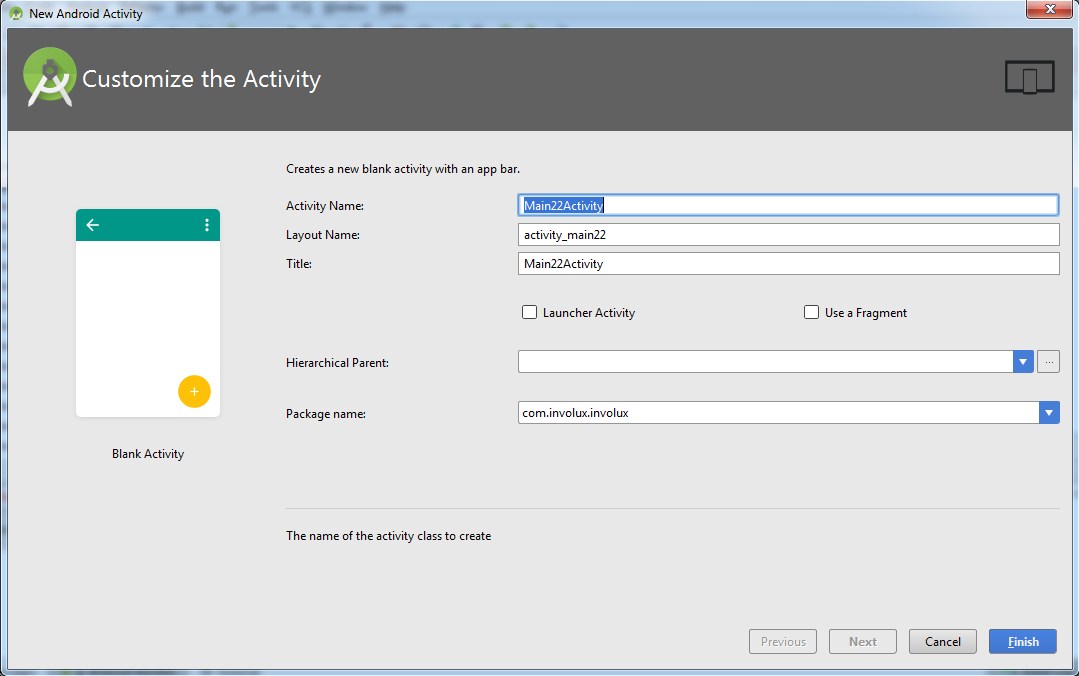
****

Рисунок 3.2 – Реквизиты новой активности

По умолчанию шаблон Blank Activity создаёт один класс активности и два файла разметки с расширением \*.xml. Один файл разметки activity\_main отвечает за отображение всей активности в целом и на второй файл разметки content\_main, который отвечает за содержимое активности, т.е. элементы формы.

Для создания активности с Navigation Drawer (боковым меню) также можно воспользоваться автоматическим созданием, но тут есть нюанс – создать активность с боковым меню можно только один раз. Это объясняется тем, что по сути создаётся не активность, а само меню, доступ к которому осуществляется путём ссылки на его ресурсы. Чтобы получить доступ каждой активности к этому меню требуется создать ещё один файл разметки, в котором и будет подключаться боковое меню.

Для обычной активности структура представляла собой слои вложенности разметки из двух уровней, т.е. существовала разметка всей активности, в которую входила разметка содержимого. Для активности с боковым меню: в разметку бокового меню входит разметка всей активности, в которую в свою очередь входит разметка содержимого:

Activity\_main.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 android:id="@+id/drawer\_layout"  
 tools:openDrawer="start"**>  
 <**include  
 layout="@layout/app\_bar\_main\_hotel\_laguna"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** />  
 <**android.support.design.widget.NavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:fitsSystemWindows="true"  
 app:headerLayout="@layout/nav\_header\_main\_company\_\_involux"  
 app:menu="@menu/activity\_main\_company\_\_involux\_drawer"** />  
</**android.support.v4.widget.DrawerLayout**>

App\_bar\_main.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:context="com.involux.involux.MainCompany\_Involux"**>  
 <**android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay"**>  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay"** />  
 </**android.support.design.widget.AppBarLayout**>  
 <**include layout="@layout/content\_main\_hotel\_\_laguna"** />  
</**android.support.design.widget.CoordinatorLayout**>

Content\_main.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/scrollView"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:background="@drawable/bc"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.involux.involux.MainHome\_Det\_Gorod"**>  
 <**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.involux.involux.MainHome\_Det\_Gorod"  
 android:maxHeight="2000dp"  
 android:maxWidth="600dp"**>  
 <**TextView  
 android:textColor="#ffffff"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAppearance="?android:attr/textAppearanceMedium"  
 android:text="LAGUNA"  
 android:id="@+id/textView"  
 android:layout\_gravity="center\_horizontal|top"  
 android:layout\_alignParentTop="true"  
 android:layout\_centerHorizontal="true"** />  
 <**TextView  
 android:textColor="#ffffff"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textAppearance="?android:attr/textAppearanceSmall"  
 android:text="  
 Безукоризненно элегантная фигурная фрезеровка на фасадах шкафов и тумб, в изголовье кровати подчеркивает индивидуальность коллекции \n  
 Кровать имеет прочные декоративные хромированные опоры с регулировкой по высоте, что позволяет без проблем производить влажную уборку помещения \n  
 Отсутствие бокового нависания во всех крышках MDF дает возможность выстраивать тумбы, столики и шкафы одинаковой высоты «под одну столешницу»\n  
 Наличие цоколя во всех шкафах и тумбах предотвращает попадание мелких предметов под мебель и предохраняет фасады от механических повреждений в процессе уборки"  
 android:id="@+id/textView2"  
 android:layout\_above="@+id/button10"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_below="@+id/textView"  
 android:textSize="12dp"  
 android:textStyle="bold"  
 android:layout\_gravity="center"  
 android:layout\_marginBottom="16dp"** />  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/sort"  
 android:layout\_gravity="left|bottom"  
 android:onClick="hotel\_laguna\_sort"  
 android:background="@drawable/hotel\_laguna\_sort"  
 android:layout\_below="@+id/textView2"  
 android:layout\_alignRight="@+id/textView2"  
 android:layout\_alignEnd="@+id/textView2"** />  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/hotel\_laguna\_foto"  
 android:id="@+id/foto"  
 android:layout\_gravity="right|bottom"  
 android:onClick="hotel\_laguna\_foto"  
 android:layout\_alignTop="@+id/sort"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginRight="16dp"** />  
 </**RelativeLayout** >  
</**ScrollView**>

Для activity\_main тег NavigationView содержит ссылку на собственную разметку в атрибуте app:headerLayout, который указывает на файл nav\_header\_main.xml (верхняя часть шторки), а также на меню в атрибуте app:menu, который ссылается на ресурс меню menu/activity\_main\_drawer.xml.

Разметка nav\_header\_main.xml и activity\_main\_drawer.xml для всего проекта являются неизменными:

Nav\_header\_main.xml:

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="@dimen/nav\_header\_height"  
 android:background="#000000"  
 android:gravity="bottom"  
 android:orientation="vertical"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 android:theme="@style/ThemeOverlay.AppCompat.Dark"**>  
<**ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:paddingTop="@dimen/nav\_header\_vertical\_spacing"  
 android:src="@drawable/header\_title"** />  
<**TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingTop="@dimen/nav\_header\_vertical\_spacing"  
 android:text="Каталог INVOLUX 2016"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"** />  
 <**TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="мебель для дома и офиса"** />  
</**LinearLayout**>

Activity\_main\_drawer.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**menu xmlns:android="http://schemas.android.com/apk/res/android"**>  
  
 <**group android:checkableBehavior="single"** >  
 <**item  
 android:id="@+id/glav"  
 android:icon="@drawable/glav"  
 android:title="Главная"** />  
 <**item  
 android:id="@+id/company"  
 android:icon="@drawable/company"  
 android:title="О компании"** />  
 <**item  
 android:id="@+id/contact"  
 android:icon="@drawable/contact"  
 android:title="Контакты"** />  
 </**group**>  
</**menu**>

**3.2 Создание элементов Activity**

Элементы активности помещаются в файл разметки content\_main. Прежде чем разместить элементы требуется определиться с типом разметки. По умолчанию файл разметки content\_main определён как RelativeLayout – элементы, помещаемые на форму активности будут иметь свободное размещение. Существует множество типов разметки таких как FrameLayout, TableLayout и т.д [2]. Выбор типа разметки зависит от информации, которая будет представляться на форме.

Если информация представляемая на форме на помещается на экране, требуется разместить разметку в контейнер ScrollView, что позволит перемещаться вверх и вниз внутри активности:

<**ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/scrollView"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="match\_parent"**>  
 <**RelativeLayout**

**android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
 </**RelativeLayout** >  
</**ScrollView**>

Создание элементов формы активности возможно как в ручном (Text) так и автоматическом режиме (Design). Ручной режим (текстовый) подразумевает полное описание элемента путём ввода кода с клавиатуры. Автоматический режим (эскизный) позволяет создать элемент через панель инструментов с установленными по умолчанию атрибутами (см. рисунок 3.3).

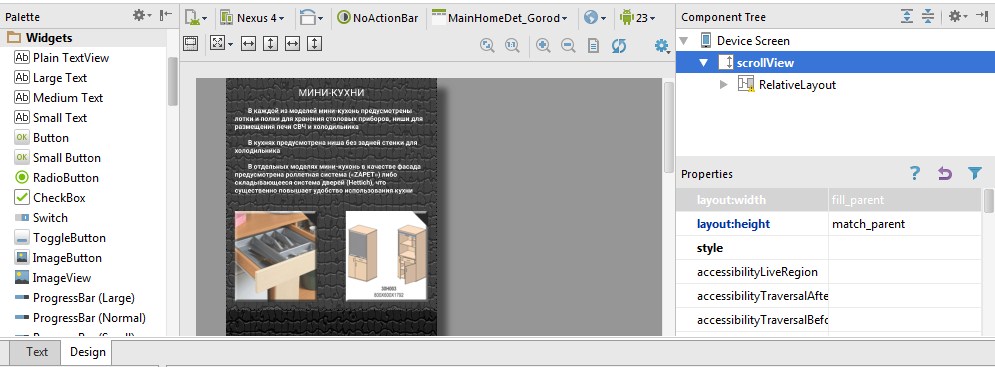


Рисунок 3.3 – Панель инструментов

В эскизном режиме также можно получить доступ к значению атрибутов элемента. Каждый элемент имеет 3 обязательных атрибута: уникальный идентификатор (id), высота элемента (layout\_width), ширина элемента (layout\_height). Пример:

<**ImageView  
 android:id="@+id/image\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"**/>

Значения атрибутов элемента могут быть как фиксированными, так и динамически изменяемыми. Фиксированные значения устанавливаются в файле разметки, динамические же задаются в классе activity. В данном случае значения высоты и ширины для кнопок рассчитывается в классе и устанавливается по завершению расчёта:

Button det\_gorod\_fotoW = (Button) findViewById(R.id.***foto***);  
Button det\_gorod\_fotoH = (Button) findViewById(R.id.***foto***);  
Button det\_gorod\_sortW = (Button) findViewById(R.id.***sort***);  
Button det\_gorod\_sortH = (Button) findViewById(R.id.***sort***);  
Display display = getWindowManager().getDefaultDisplay();  
DisplayMetrics metricsB = **new** DisplayMetrics();  
display.getMetrics(metricsB);  
  
det\_gorod\_fotoW.getLayoutParams().**width**= (**int**) (metricsB.**widthPixels**/2.4);  
det\_gorod\_fotoH.getLayoutParams().**height**=(**int**) (metricsB.**heightPixels**/3.8);  
det\_gorod\_sortW.getLayoutParams().**width**= (**int**) (metricsB.**widthPixels**/2.4);  
det\_gorod\_sortH.getLayoutParams().**height**=(**int**) (metricsB.**heightPixels**/3.8);

Такой расчет нужен для того чтобы кнопки корректно отображались на всех устройствах Android с различными размерами экрана. Т.е. при открытии или переходе на активность приложение получает разрешение экрана и путём тестовых расчётов размера кнопок на форме, рассчитываются и устанавливаются значения атрибутов высота и ширина для кнопок. Коэффициент для расчета размеров кнопок получается экспериментальным путём, т.е. математически рассчитать размер кнопок для всех размерностей экранов весьма затруднительно. Для этого берётся несколько устройств с различным разрешением и путём увеличения и уменьшения коэффициента определяется оптимальный размер. Размер текст не принципиально рассчитывать динамически, т.к. для TextView установлен размер равный содержимому.

* 1. **Создание переходов между Activity**

Переходы между активностями осуществляются при помощи кнопок. У кнопок есть относительно новый способ, специально разработанный для Android - использовать атрибут onClick [3] (на панели свойств отображается как On Click). Следует написать обработку для данного метода, т.е. прописать в классе активности придуманное имя метода, который будет обрабатывать нажатие. Метод должен быть открытым (public). Для запуска нового экрана необходимо создать экземпляр класса Intent (намерения) и указать в первом параметре текущий класс, а во втором - класс для перехода. После этого вызывается метод startActivity(), который и запускает новый экран:

**public void** gorod\_foto(View view) {  
 Intent intent1 = **new** Intent(MainHome\_Det\_Gorod.**this**, MainDet\_Gorod\_foto.**class**);  
 startActivity(intent1); }

Актрибут кнопки для данной обработки нажатия выглядит так:

**android:onClick="gorod\_foto"**

Также в проекте существуют активности доступ, к которым осуществляется с помощью Navigation Drawer. Поэтому в классе активности для бокового меню описывается метод для обработки выбора элементов:

**public boolean** onNavigationItemSelected(MenuItem item) {  
 *// Handle navigation view item clicks here.* **int** id = item.getItemId();  
 **if** (id == R.id.***glav***) {  
 {  
 Intent intent1 = **new** Intent(MainHome\_Det\_Gorod.**this**, MainActivity.**class**);  
 startActivity(intent1); }  
 } **else if** (id == R.id.***company***) {  
 {  
 Intent intent1 = **new** Intent(MainHome\_Det\_Gorod.**this**, MainCompany\_Involux.**class**);  
 startActivity(intent1); }  
 } **else if** (id == R.id.***contact***) {  
 Intent intent1 = **new** Intent(MainHome\_Det\_Gorod.**this**, Main\_Contact.**class**);  
 startActivity(intent1); }  
 DrawerLayout drawer = (DrawerLayout) findViewById(R.id.***drawer\_layout***);  
 drawer.closeDrawer(GravityCompat.***START***);  
 **return true**;  
}

Как видно также используются намерения (Intent) для перехода между активностями.

* 1. **Создание галереи изображений**

Для того чтобы добиться эффекта слайдинга изображений требуется элемент View Pager, для которого отдельно создаётся файл разметки содержащий TextView и ImageView – элементы которые будут динамически изменяться в процессе работы программы [4]. По сути, отдельно созданный файл разметки является контейнером для хранения и отображения информации.

Также следует создать отдельный класс-помощник, который и будет передавать выполняемые им действия в основную активность. Но для начала следует предварительная обработка изображений, чтобы минимизировать ресурсы затрачиваемые устройство для их отображения. Для этого следует конвертировать изображения в формат \*.gif (для web и устройств). Данная процедура уменьшит физический размер файлов и не ухудшит качество, что идеально подходит для данного приложения. Далее следует загрузить изображения на фото-хостинг, для последующего отображения в приложении (см. рисунок3.4-3.5).

Загрузив изображения в альбомы – выгружаем ссылки на них и помещаем в массив класса слайдинга:

**private** String[] **image\_resources**={  
 **"http://i.imgur.com/wY6NrDq.gif"**, **"http://i.imgur.com/r06lVUI.gif"**,  
 **"http://i.imgur.com/cYi2PGB.gif"**, **"http://i.imgur.com/5zLalQo.gif"**,  
 **"http://i.imgur.com/6AZgFdP.gif"**, **"http://i.imgur.com/mK92B1z.gif"**,  
 **"http://i.imgur.com/tDwGIf7.gif"**, **"http://i.imgur.com/hy3fvf5.gif"**,  
 **"http://i.imgur.com/44jktk7.gif"**, **"http://i.imgur.com/E1mpzgC.gif"**};

Для позиционирования изображений определяется длина массива. Это нужно для того чтобы пользователь мог знать количество изображений, а также порядковый номер изображения которое в данный момент отображается. Информация об этом будет загружаться в TextView.

Для загрузки изображения из интернета следует подключить библиотеку Picasso помогающую это осуществить. Подключить библиотеку можно прописав в build.gradle (Module:app):

dependencies {  
 compile **'com.squareup.picasso:picasso:2.5.2'** compile **'com.github.bumptech.glide:glide:3.7.0'**}

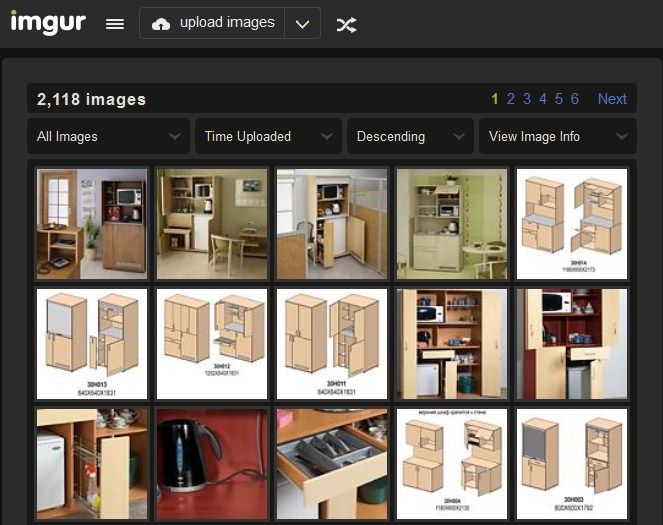


Рисунок 3.4 – Загруженные изображения на фото-хостинг

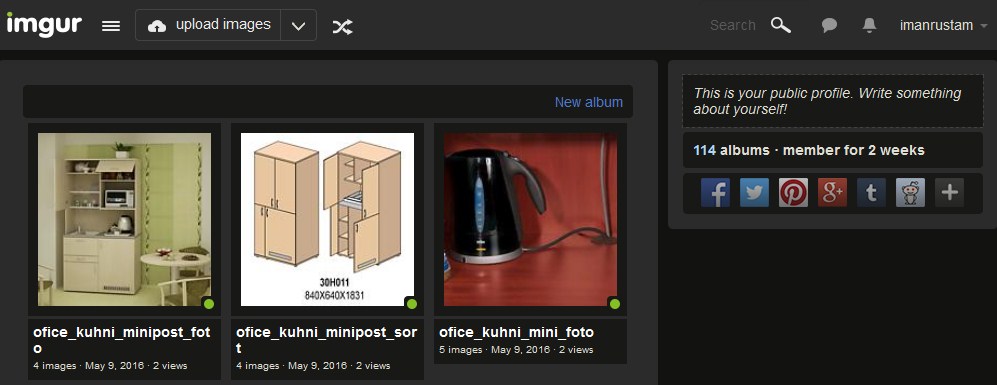


Рисунок 3.5 – Альбомы изображений фото-хостинга

Библиотека Picasso [5], предназначенная для асинхронной загрузки изображений из сети, ресурсов или файловой системы, их кэширования и отображения. Чтобы воспользоваться данной библиотекой, следует прописать в манифесте зависимость, или же подключить библиотеку. Gradle сама скачает и установит библиотеку в проект.

Также следует включить разрешение для использования приложением интернета:

<**uses-permission android:name="android.permission.INTERNET"** />

Использование библиотеки Picasso упрощает загрузку изображений из интернета, а также автоматически кеширует их:

Picasso.*with*(**ctx**)  
 .load(**image\_resources**[position])  
 .placeholder(R.drawable.***loading***)  
 .error(R.drawable.***down\_error***)  
 .into(imageView);

Принцип использования библиотеки очень прост: указывается адрес картинки (url), заглушка (placeholder), заглушка для ошибки после трёх неудачных попыток загрузки (error) и в методе into() указывается компонент ImageView, в который загружается изображение. В данном случае в качестве адреса изображения используется массив адресов с указанием позиции в массиве. Так как массив не циклический по умолчанию загружается первое из массива изображение, а далее с шагом один по порядку загружаются остальные изображения путём перехода "[Swype](http://ru.wikipedia.org/wiki/Swype)" - провести пальцем влево или вправо, не отрывая его от экрана (см. рисунок 3.5).

Использование библиотеки Picasso позволяет экономить ресурсы на подключение к интернету. Библиотека позволяет хранить загруженные изображения в кэш памяти до закрытия приложения.

Класс-помощник предаёт информацию в основной класс и поэтому изменения при переходе Swype будут, происходит только тех элементов, которые входят в состав класса-помощника.

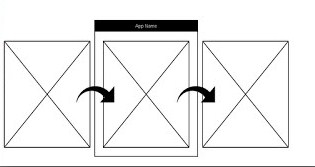


Рисунок 3.7 – Принцип работы Swype

4 РЕАЛИЗАЦИЯ И ИСПЫТАНИЕ АСОИ

Программная система представления информации о продукции ЗАО «Инволюкс» реализована с помощью языка высокого уровня Java и среды разработки Android Studio. Испытание системы заключается в проверке соответствия возможностей созданной системы требованиям, заданным в техническом задании. Для демонстрации проверки системы рассмотрим последовательности следующих тестов:

* загрузка и установка приложения;
* реализация и работоспособность Splash Screen;
* реализация и работоспособность Navigation Drawer;
* реализация и работоспособность кнопок;
* реализация и работоспособность просмотра коллекций.

4.1 Загрузка и установка приложения

Загрузка приложения на устройство происходит или в ручном, или в полуавтоматическом режиме.

Загрузка в ручном режиме подразумевает копирование из внешних источников установочного файла с расширением \*.apk, сгенерированного с помощью среды разработки Android Studio. Минус данного способа установки в том, что по умолчанию установка приложения на устройство не через сервис Play Market, запрещена в настройках как «Неизвестные источники». Найти и снять/установить галочку с данного ограничения можно через настройки телефона, в пункте меню «Безопасность».

Загрузка в полуавтоматическом режиме означает использования сервиса Play Market - магазин приложений, игр, книг, музыки и фильмов компании Google и других компаний, позволяющий владельцам устройств с операционной системой Android устанавливать и приобретать различные приложения. Учётная запись разработчика, которая даёт возможность публиковать приложения, стоит $25. Платные приложения могут публиковать разработчики не из всех стран.

Стоит заметить, что при разработке приложения отладка возможна как на виртуальном устройстве AVD (Android Virtual Device), так и на реальном устройстве с помощью USB кабеля и соответствующих драйверов. Отладка по USB позволяет использовать службу отладки приложений и устройств на ОС Android: проверить, как работает приложение и система, какие возникали сбои.

После того как приложение будет успешно установлено, в меню «Приложения» появится ярлык с иконкой и соответствующей подписью (см. рис 4.1).

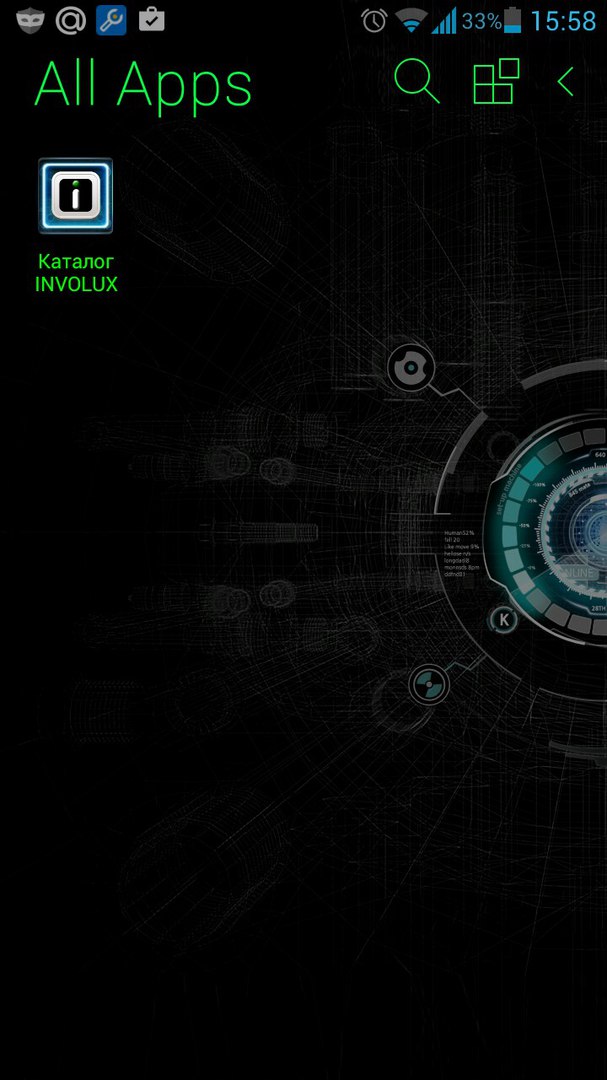


Рисунок 4.1 – Ярлык приложения

4.2 Реализация и работоспособность Splash Screen

Как говорилось ранее Splash Screen - это отображение какого-либо изображения, а также дополнительной информации о компании и приложении, в то время, пока загружается главная форма приложения. Splash Screen по сути является активностью (Activity) которая появляется при запуске приложения, затем после определённого продолжительности времени, в данном случае это 3 секунды, загружается основная форма приложения (см. рисунок 4.2).



Рисунок 4.2 – Splash Screen

Как видно из рисунка 4.2, данная Activity имеет:

* название приложения;
* логотип компании;
* информацию о приложении.

Рассмотри более детально каждый пункт.

Название приложения берёт своё начало из файла AndroidManifest.xml - предоставляет подробную информацию о приложении. В каждом Android существует этот файл, так как он является обязательным для создания приложения на ОС Android.

В AndroidManifest.xml можно настроить следующие возможности:

* указать имя Java-пакета приложения, который служит уникальным идентификатором;
* описать компоненты приложения, службы;
* указать список необходимых разрешений для обращения к защищенным частям API и взаимодействия с другими приложениями;
* разрешить доступ к ресурсам, которые сторонние приложения обязаны иметь для взаимодействия с компонентами данного приложения;
* указать минимальный и максимальный уровень API Android, необходимый для работы приложения;

Именно имя приложения, описанное в манифесте, служит надписью в верхней части Splash Screen.

Логотип компании это изображение, помещённое в ImageView и привязанное полями к центру экрана. ImageView является базовым элементом-контейнером для использования графики.

Информация о приложении содержит сведения о текущем состоянии приложения:

* «BETA – Каталог продукции» - интенсивное использование почти готовой версии продукта с целью выявления максимального числа ошибок в его работе для их последующего устранения перед окончательным выходом (релизом) продукта на рынок, к массовому потребителю.
* «Copyright 2016 © Imanov Rustam | BSTU. All right reserved». - Авторское право на приложение и соответственно имя автора. Целью копирайта является охрана авторского права. Правообладатель, разместивший в своих материалах знак Copyright, защищает себя от актов плагиата и пиратства. Регистрировать знак охраны авторских прав не нужно, достаточно указать его в своих работах и он автоматически вступит в силу, так как является утвержденным международным стандартом охраны прав авторской собственности, принятым Всемирной (Женевской) Конвенцией более полувека назад. Существует ряд правил касательно того, как грамотно указать знак копирайта в своем материала. Во первых ставиться сам значок копирайта – «©» или слово Copyright, далее идет имя создателя или адрес ресурса, которому принадлежат авторские права, и после указывается дата создания данного произведения, обычно просто год. Соблюдение правил оформления копирайта является гарантом его юридической силы.
* «Version 2.23b» - версификация приложения. Жизненный цикл успешной компьютерной программы может быть очень долгим; изменения в программе бывают разными — от исправления ошибки до полного переписывания. В большинстве случаев название программы остаётся тем же, изменяется подназвание — так называемая версия. Расшифровывается как вторая рабочая версия с найденными и исправленными 23 корректировками. Буква «b» означает что эта версия активно тестируется.

4.3 Реализация и работоспособность Navigation Drawer

Как говорилось ранее Navigation Drawer это панель, которая отображает основные опции навигации приложения по левому или правому краю экрана (см. рисунок 4.3)

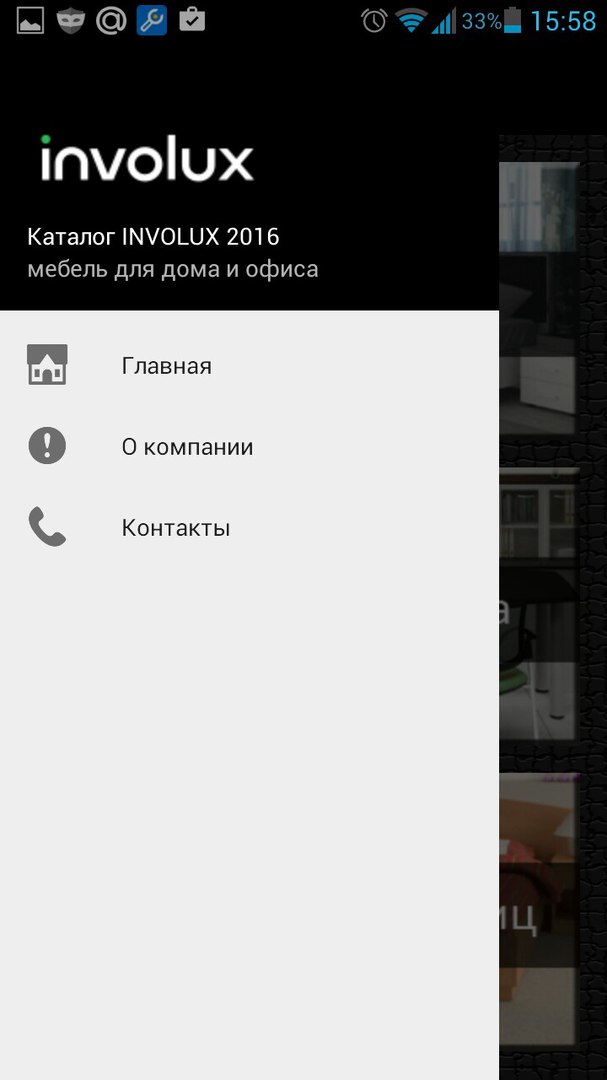


Рисунок 4.3 – Боковое меню Navigation Drawer

Тег **NavigationView** содержит ссылку на собственную разметку в атрибуте **app:headerLayout**, который указывает на файл **nav\_header\_main.xml** (верхняя часть шторки), а также на меню в атрибуте **app:menu**, который ссылается на ресурс меню **menu/activity\_main\_drawer.xml**. За выдвигающую шторку отвечает элемент **NavigationView**, который входит последним в контейнере **DrawerLayout** и представляет собой навигационное меню. А перед меню находится вставка **include**, указывающая на разметку **app\_bar\_main.xml**, поэтому нужно объявить пользовательский интерфейс с объектом DrawerLayout в качестве корневого вида макета. Внутри DrawerLayout, добавьте один вид, который содержит главный контент для экрана (ваш основной макет, когда навигационный ящик скрыт) и другой вид, который содержит контент навигационного ящика.

Реализация разметки шторки **nav\_header\_main.xml**. Разметка состоит из **ImageView** и двух **TextView**, размещённых в контейнере **LinearLayout**. Фон контейнера определён в ресурсе **drawable/side\_nav\_bar.xml** и представляет собой цвет.

Реализация и работоспособность ресурса навигационного меню **res/menu/activity\_main\_drawer.xml**. Принцип создания элементов меню стандартен. Каждый пункт меню представляет собой тег item с указанием значка и текста. Для группировки используется элемент group. Поведение элементов меню в группе регулируется атрибутом android:checkableBehavior. В примере используется значение single - при нажатии на пункт меню, он останется выделенным (принцип переключателя RadioButton). Всего доступно три варианта:

* single – можно выбрать один элемент группы (переключатель);
* all - можно выбрать все элементы группы (флажок);
* none – элементы не выбираются.

Реализация работоспособности бокового меню. В классе активности реализуется интерфейс OnNavigationItemSelectedListener с его методом onNavigationItemSelected().Логика кода ничем не отличается о работы с обычным меню. Определяется идентификатор выбранного пункта и далее вызывается метод **closeDrawer()** для закрытия шторки.

Обработка событий кликов Navigation Drawer. Когда пользователь выбирает элемент в списке навигационного ящика, система вызывает onItemClick () слушателя OnItemClickListener определенного в setOnItemClickListener ().

Слушание событий открытия и закрытия навигационного ящика. Для прослушивания событий открытия и закрытия навигационного ящика, вызовите метод setDrawerListener () для DrawerLayout и передайте ему реализацию DrawerLayout.DrawerListener. Этот интерфейс обеспечивает методы обратного вызова, такие как onDrawerOpened () и onDrawerClosed ().

Обеспечение надлежащей навигации назад. Навигация назад это как пользователи перемещаются назад через историю экранов, которые они ранее посещали. Все Android устройства обеспечивают кнопку «Назад» для этого типа навигации.

Почти во всех ситуациях система поддерживает стек активностей, в то время как пользователь осуществляет навигацию приложения. Это позволяет системе правильно перемещаться назад, когда пользователь нажимает кнопку «Назад».

С бокового меню можно перейти: на главную форму, на форму «О компании» (см. рисунок 4.4), на форму «Контакты» (см. рисунок 4.5).

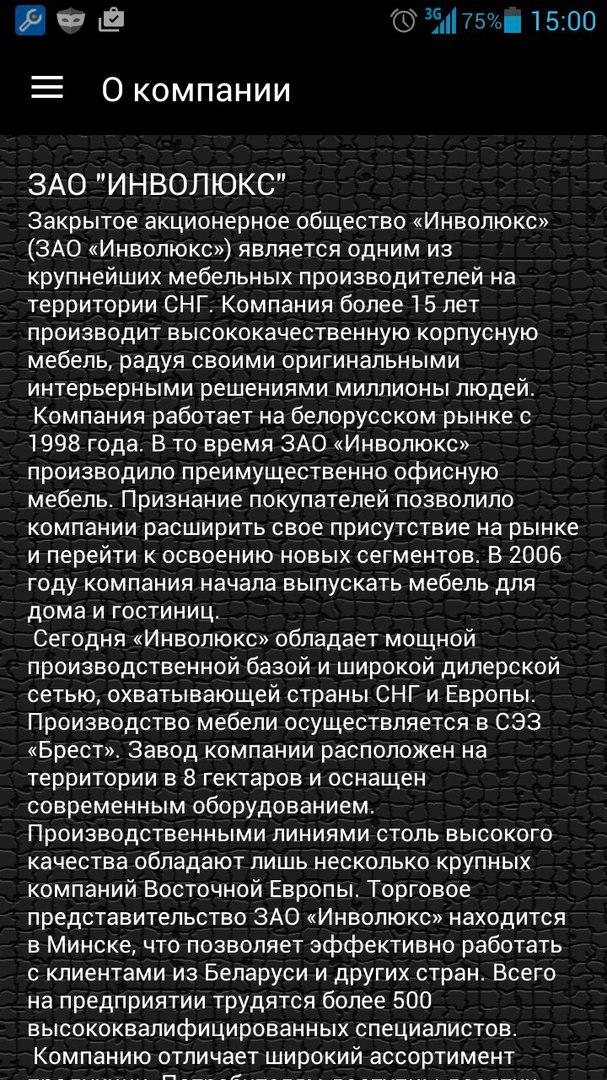


Рисунок 4.4 – Форма «О компании»

На форму «О компании» и «Контакты» можно перейти только через боковое меню.

Форма «О компании» - представляет собой текст, размещённый в TextView. Содержание этого текста гласит историю предприятия, вид деятельности, цели и достижения.

Форма «Контакты» - это тоже текст, размещенный в TextView, но особенность его состоит в том, что присутствует свойство для автоматического определения гиперссылок AutoLink= «all». По нажатию на выделенный текст происходит действие, автоматически определяемое устройством как URL адрес или же номер телефона, и соответственно переходи или в браузер или в набор номера с уже сохранёнными данными для ввода информации. Стоит заметить, что E-mail также автоматически определяется и нажав на выделенный почтовый адрес, можно отправить электронное письмо, либо через встроенный сервис или же через предварительно установленные приложения для работы с почтовыми ящиками – система предоставляет выбор.

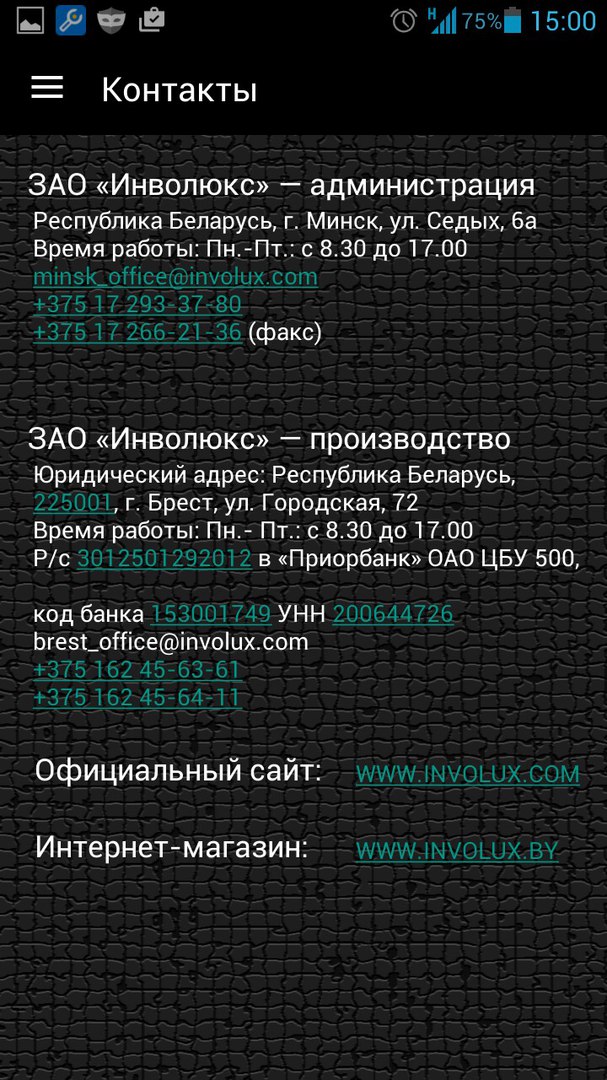


Рисунок 4.5 – Форма «Контакты»

* 1. Реализация и работоспособность кнопок

Кнопка – один из самых распространённых элементов управления в программировании. В Android для кнопки используется класс android.widget.Button. На кнопке располагается текст и на кнопку нужно нажать, чтобы получить результат. В данном случае у кнопок отсутствует текст, присутствует атрибут background и используется изображение для кнопки.

Чтобы кнопка имитировала нажатие изображением служит Selector – элемент управления различных состояний. В данном случае кнопка должна находится в двух состояниях: pressed и default, т.е. нажато и по умолчанию (см.рисунок4.6). Selector по сути это файл разметки XML в котором описываются состояния. Находится этот файл должен в каталоге (классе) drawable. Класс Drawable является базовым классом для всех классов работы с графикой.

Рисунок 4.6 – Принцип работы Selector

Кнопкам как и впринципе всем элементам форм присущи размеры. Чтобы кнопки отображались корректно на всех устройствах с различными разрешениями экрана и плотностью следует динамически рассчитывать размеры кнопок исходя из пропорциональных размеров экрана.

* 1. Реализация и работоспособность просмотра коллекций

Размер экрана не всегда позволяет отобразить всю необходимую информацию внутри одной Activity. Хорошим примером является стандартное приложение «Галерея», где можно просматривать любое количество изображений «по одному», листая их с помощью жеста «[Swype](http://ru.wikipedia.org/wiki/Swype)».

Чтобы реализовать «слайдинг» изображений надо воспользоваться классом View Pager. ViewPager [6] позволяет организовать удобный и красивый просмотр данных с возможностью перелистывания влево-вправо.

Количество страниц для каждой коллекции определяется заданным массивом ссылок на изображения размещённые в интернете. Также применяется счётчик элементов массива. Загрузка же изображений для просмотра происходит через интернет путём использования библиотеки Picasso.

Принцип работы библиотеки прост:

* указывается адрес картинки (url);
* заглушку (placeholder);
* заглушку для ошибки после трёх неудачных попыток загрузки (error);
* в методе into() указывается компонент ImageView, в который загружается изображение.

В приложении в качестве адреса картинки указан массив, с адресами на изображения. В сочетании с ViewPager и Picasso происходит то, что с каждым движением Swype происходит переход по массиву с шагом по умолчанию равным 1. Переход по массиву не является циклическим, т.е. с последнего изображения нельзя перейти к первому и наоборот.

Как видно из рисунка 4.7 включена передача данных или же доступ к сети интернет через wi-fi и изображение благополучно загрузилось. Для того чтобы пользователь знал что происходит загрузка, а также если произошла неудачная попытка загрузить изображение, используются заглушки. Заглушка загрузки изображения и заглушка неудачной загрузки для всех изображений остаётся одинаковой (см. рисунок 4.8).

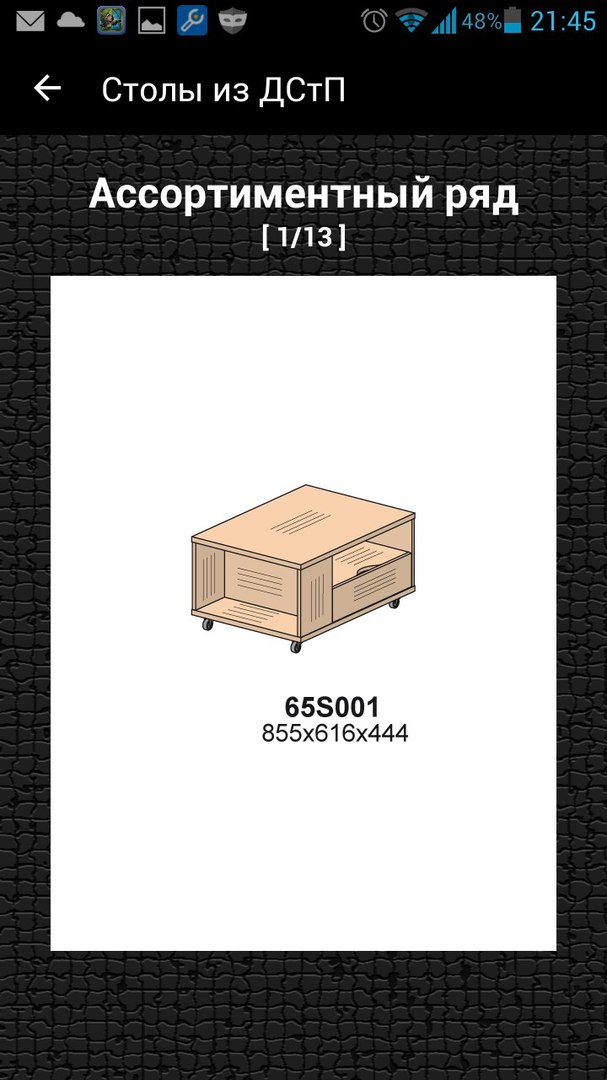


Рисунок 4.7 – Слайдинг изображений

Рисунок 4.8 – Работоспособность заглушек

Как видно из рисунка 4.8 прежде чем изображения загрузится появляется сообщение «Идёт загрузка...» - это сообщает пользователю что запущен некий процесс в данном случае загрузка изображения. Также следует заметить что если будет отключён доступ к интернету появится сообщении «Ошибка. Подключитесь к сети». Ошибка конкретизированна, т.к. отсутствуют другие факторы которые могут препятствовать загрузке изображения.

Место хранения любезно предоставляет онлайн-сервис загрузки, хранения и обмена фото-изображений imgur ([www.imgur.com](http://www.imgur.com)). Данный сервис предоставляет безграничный, надёжный и высокоскоростной доступ к изображениям в сети интернет.

**5 РАСЧЕТ ЭКОНОМИЧЕСКИХ ПОКАЗАТЕЛЕЙ**

## 5.1 Расчет объема функций программного обеспечения

В данном дипломном проекте разрабатывается программное обеспечение (ПО), которое представляет собой разработку приложения, обеспечивающее сокращение трудозатрат по ведению информации при решении комплекса задач по организации и ведению документов при обслуживании юридических лиц на предприятии.

Разработка программного средства (ПС) предусматривает проведение практически всех стадий проектирования (техническое задание, эскизный проект, технический проект, рабочий проект), относится ко второй группе сложности.

Среда разработки Java. Данное ПО является развитием определенного параметрического ряда ПО, применяемое на существующих типах ЭВМ и существующей ОС.

Расчет общего объема ПО (количества строк исходного кода (LOC)) предполагает определение объема по каждой функции, реализуемой программой по формуле 5.1:

|  |  |
| --- | --- |
|  | (5.1) |

где – общий объем ПС;

– объем отдельной функции ПС;

– общее число функций.

На основании информации о функциях разрабатываемого ПО по каталогу функций определим общий объем ПО. В зависимости от организационных и технологических условий, в которых разрабатывается ПО, был скорректирован объем на основе экспертных оценок.

Уточненный объем ПО определяется по формуле 5.2:

|  |  |
| --- | --- |
|  | (5.2) |

где – уточненный объем ПС;

– уточненный объем отдельной функции ПС;

– общее число функций.

Перечень и объем функций программного обеспечения приведен в таблице 5.1.

Таблица 5.1 – Перечень и объем функций программного обеспечения

|  |  |  |  |
| --- | --- | --- | --- |
| Код функции | Наименование (содержание) функции | Объем функции строк исходного кода (LOC) | |
| По каталогу Vi | Уточненный Vуi |
| 1 | 2 | 3 | 4 |
| 101 | Организация ввода информации | 130 | 120 |
| 102 | Контроль, предварительная обработка и ввод информации | 490 | 460 |
| 107 | Организация ввода/вывода информации в интерактивном режиме | 280 | 245 |
| 301 | Формирование последовательного файла | 590 | 485 |
| 302 | Автоматическая сортировка файла | 890 | 850 |
| 303 | Обработка файлов | 1 050 | 1 000 |
| 304 | Управление файлами | 5 240 | 5 220 |
| 305 | Формирование файла | 2 130 | 1 800 |
| Итого: |  | 10 800 | 10 180 |

LOC - lines of code - количество строк кода - мера сложности программного обеспечения.

Учитывая информацию, указанную в таблице 5.1, о функциях разрабатываемого программного обеспечения, уточненный общий объем ПО составил 10180 строк исходного кода (LOC) вместо предполагаемого количества строк 10800.

## 5.2 Расчет заработной платы разработчиков программного обеспечения

Основной статьей расходов на создание ПП является заработная плата проекта (основная и дополнительная) разработчиков (исполнителей) , в число которых принято включать инженеров-программистов, руководителей проекта, системных архитекторов, дизайнеров, разработчиков баз данных, Web-мастеров и других специалистов, необходимых для решения специальных задач в команде.

Расчёт заработной платы разработчиков ПП начинается с определения:

* продолжительности времени разработки с учётом сложности, новизны ПП и фактически затраченного времени установим равным 2 месяца.
* количества разработчиков ПП. В данном дипломном проекте 2 разработчика: инженер-программист 1 категории, инженер-программист 2 категории.

Заработная плата разработчиков определяется как сумма основной и дополнительной заработной платы всех исполнителей.

Основная заработная плата каждого исполнителя определяется по формуле 5.3:

|  |  |
| --- | --- |
|  | (5.3) |

где – месячная тарифная ставка 1 разряда рабочего (с 1 апреля 2016 года на предприятии – 350 000 рублей);

– тарифный коэффициент согласно разряду исполнителя;

21,3 – среднее количество рабочих дней в месяце;

– фонд рабочего времени исполнителя (продолжительность разработки ПП, дни);

– коэффициент премий, .

3 818 369 (бел. руб.),

3 336 667 (бел. руб.),

Дополнительная заработная плата каждого исполнителя рассчитывается от основной заработной платы по формуле: 5.4

|  |  |
| --- | --- |
|  | (5.4) |

Результаты вычислений внесем в таблицу 5.2.

Таблица 5.2 – Расчет заработной платы

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № п/п | Категории работников | Разряд | Тарифный коэффициент (Кт) | Фэф, дн. | Тарифная ставка  I разряда | Коэффициент премирования (Кпр) | Норматив доп. з/пл (НЗПд) | Заработная плата, тыс.руб. | | |
| Основная | Дополнительная | Всего |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | Инженер-программист 1 категории | 14 | 3,25 | 5 | 350 000 | 1,3 | 30 | 3 818 369 | 1 145 511 | 4 963 880 |
| 2 | Инженер-программист 2 категории | 12 | 2,84 | 5 | 350 000 | 1,3 | 30 | 3 336 667 | 1 001 000 | 4 337 667 |
|  | **ИТОГО** | | | | | | | 7 155 036 | 2 146 511 | 9 301 547 |

## 5.3 Расчет себестоимости программного обеспечения

Стоимостная оценка ПС у разработчиков предполагает составление сметы затрат, которая включает в себя следующие статьи:

* основная заработная плата исполнителей ;
* дополнительная заработная плата исполнителей ;
* отчисления на социальные нужды ;
* материалы и комплектующие ;
* спецоборудование ;
* машинное время ;
* расходы на научные командировки ;
* прочие прямые расходы ;
* накладные расходы ;
* затраты на сопровождение и адаптацию программного средства и .

Отчисления на социальные нужды определяются в соответствии с действующими законодательными актами по нормативу в процентном отношении к фонду основной и дополнительной заработной платы исполнителей. В 2016 году норматив отчислений в ФСЗН составляет 34%, отчислений по обязательному страхованию составляет 0,6%. Расходы на социальные нужды рассчитаем по формуле 5.5:

|  |  |
| --- | --- |
|  | (5.5) |

,

Расходы по статье «Материалы и комплектующие изделия» определяются на основании сметы затрат, разрабатываемой на программное средство с учетом действующих нормативов.

Норма расхода материалов (НМ) определяется в расчете на 100 машинных команд.

Сумма затрат на расходные материалы (бел. руб.) определяем по формуле 5.6:

|  |  |
| --- | --- |
|  | (5.6) |

где – норма расхода материалов в расчете на 100 строк исходного кода ПП,

– уточненный общий объем функций строк исходного кода (LOC).

Расходы по статье «Спецоборудование» включают затраты средств на приобретение вспомогательных специального назначения технических и программных средств, необходимых для разработки конкретного программного средства, включая расходы на их проектирование, изготовление, отладку, установку и эксплуатацию.

Данная статья включается в смету расходов и на разработку ПС в том случае, когда приобретаются специальное оборудование или специальные программы, предназначенные для разработки и создания только данного программного средства. Поскольку в данном дипломном проекте приобретение какого-либо оборудования не предусматривалось, данная статься не рассчитывается.

Расходы по статье «Машинное время» включает оплату машинного времени, необходимого для разработки и отладки ПС.

Машинное время определяется по нормативам в машино-часах на 100 команд машинного времени в зависимости от характера решаемых задач и типа ПЭВМ по формуле 5.7:

|  |  |
| --- | --- |
|  | (5.7) |

где – цена одного машино-часа,

– уточненный общий объем функций строк исходного кода (LOC);

– норматив расхода машинного времени на отладку 100 строк кода, машино-часов. Принимаем в размере 0,9.

Расходы по статье «Научные командировки» на конкретное программное средство определяется по нормативу в процентах к основной заработной плате. Поскольку в данном проекте научные командировки не предусмотрены, данная статья не рассчитывается.

Расходы по статье «Прочие затраты» на конкретное программное средство включают затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы. Определяются по нормативу в процентах к основной заработной плате. Так как никакая специальная научно-техническая информация и специальная литература не приобреталась, то данная статья не рассчитывается.

Затраты по статье «Накладные расходы» связаны с содержанием аппарата управления, вспомогательных хозяйств и экспериментальных производств, общехозяйственными расходами.

Накладные расходы относятся на конкретное программное средство по нормативу в процентном отношении к основной заработной плате исполнителей по – определяется по формуле 5.8:

|  |  |
| --- | --- |
|  | (5.8) |

где ННР – норматив накладных расходов. В данном дипломном проекте норматив накладных расходов примем 25 %.

Сумма вышеперечисленных расходов по статьям на ПС определяет производственную себестоимость ПС и служит исходной базой для расчёта затрат на освоение и сопровождение ПС – определяется по формуле 5.9:

|  |  |
| --- | --- |
|  | (5.9) |

Организация-разработчик участвует в освоении ПП и несёт соответствующие затраты, на которые составляется смета, оплачиваемая заказчиком по договору. Затраты на освоение определяются по установленному нормативу от производственной себестоимости по формуле 5.10:

|  |  |
| --- | --- |
|  | (5.10) |

где – норматив отчислений, %. В данном дипломном проекте принимается равным 10%.

Организация-разработчик осуществляет сопровождение ПП и несёт расходы, которые оплачиваются заказчиком в соответствии с договором и сметой на сопровождение. Затраты на сопровождение определяются по установленному нормативу от производственной себестоимости по формуле 5.11:

|  |  |
| --- | --- |
|  | (5.11) |

где – норматив отчислений, %. В данном дипломном проекте принимается равным 10%.

Полная себестоимость разработки программного средства рассчитывается как сумма расходов по всем статьям и определяется по формуле 5.12:

|  |  |
| --- | --- |
|  | (5.12) |

Выполним расчет себестоимости в табличной форме (см. таблицу 5.3).

Таблица 5.3 – Расчет себестоимости ПО

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Наименование статей затрат | Норматив | Расчетная формула | Сумма затрат, руб. |
| A | B | C | D | E |
| 1 | Заработная плата, всего  в т.ч. | - | - | 9 301 547 |
| 1.1 | Основная заработная плата | - | - | 7 155 036 |
| 1.2 | Дополнительная заработная плата | - | - | 2 146 511 |
| 2 | Отчисления на социальные нужды | 34,6% |  | 3 218 335 |
| 3 | Материалы | - | формула 4.6 | 75 600 |
| 4 | Спецоборудование | не применялось | | |
| 5 | Машинное время | - | формула 4.7 | 1 944 000 |
| 6 | Научные командировки | не планировались | | |
| 7 | Прочие затраты | не применялись | | |
| 8 | Накладные расходы | 25% |  | 1 788 759 |
| 9 | Производственная себестоимость | - |  | 16 328 241 |
| 10 | Расходы на освоение | 10% |  | 1 632 824 |
| 11 | Расходы на сопровождение | 10% |  | 1 632 824 |
| 12 | Полная себестоимость | - | 9Е+ 10Е+11Е | 19 593 889 |

Полная себестоимость программного продукта составляет 18 194 209 рублей.

## 5.4 Расчет отпускной цены программного обеспечения

Отпускная цена предприятия – это цена, при которой обеспечивается возмещение текущих затрат производства и получение прибыли.

Для определения цены ПП необходимо рассчитать плановую прибыль, которая рассчитывается по формуле 5.13:

|  |  |
| --- | --- |
|  | (5.13) |

где – полная себестоимость программного продукта;

R – уровень рентабельности ПП. В данном дипломном проекте уровень рентабельности принимается равным 30%.

После расчета прибыли от реализации определяется прогнозируемая цена ПП без налогов по формуле 5.14:

|  |  |
| --- | --- |
|  | (5.14) |

где – полная себестоимость программного продукта;

П – плановая прибыль от реализации ПО.

Отпускная цена (цена реализации) ПП включает налог на добавленную стоимость и рассчитывается по формуле 5.15:

|  |  |
| --- | --- |
|  | (5.15) |

где – полная себестоимость программного продукта;

П – плановая прибыль от реализации ПО;

НДС – налог на добавленную стоимость рассчитывается по формуле (5.16):

|  |  |
| --- | --- |
|  | (5.16) |

где – прогнозируемая цена;

НДС – налог на добавленную стоимость, в настоящее время составляет 20%.

Прибыль от реализации ПП за вычетом налога на прибыль является чистой прибылью, остается организации-разработчику и представляет собой экономический эффект от создания нового программного продукта и вычисляется по формуле 5.17:

|  |  |
| --- | --- |
|  | (5.17) |

где – налог на прибыль по ставке 18%.

Рассчитывается по формуле (5.18):

|  |  |
| --- | --- |
|  | (5.18) |

Все расчёты цены и прибыли по программному продукту сведены в таблицу 5.4.

Таблица 5.4 – Расчет цены и прибыли по ПП

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Наименование статей затрат | Норматив | Расчетная формула | Сумма затрат, руб. |
| A | B | C | D | E |
| 1 | Полная себестоимость | - | - | 19 593 889 |
| 2 | Прибыль | 30% |  | 5 878 167 |
| 3 | Цена без НДС | - | 1Е+2Е | 25 472 056 |
| 4 | Налог на добавленную стоимость | 20% |  | 3 918 778 |
| 5 | Отпускная цена | - | 3Е+4Е | 29 390 834 |
| 6 | Налог на прибыль | 18% |  | 1 058 070 |
| 7 | Чистая прибыль | - | 2Е - 6Е | 4 820 097 |

Итак, определены основные экономические показатели:

Полная себестоимость – 19 593 889 бел. руб.

Прогнозируемая цена – 29 390 834 бел. руб.

Чистая прибыль – 4 820 097 бел. руб.

Согласно различным источникам, текущая рыночная цена на подобный программный продукт в Республике Беларусь колеблется в диапазоне от 25 000 000 бел.руб. до 35 000 000 бел.руб. Таким образом, рассчитанная отпускная цена на программный продукт, разрабатываемый в рамках данного дипломного проекта, является конкурентоспособной, в связи с чем, создание данного программного продукта будет экономически целесообразным.

6 ОХРАНА ТРУДА И ЭКОЛОГИЧЕСКАЯ БЕЗОПАСНОСТЬ

Охрана труда - система законодательных актов, социально-экономических, организационных, технических, гигиенических и лечебно-профилактических мероприятий и средств, обеспечивающих безопасность, сохранение здоровья и работоспособности человека в процессе труда.

Научно-технический прогресс внес серьезные изменения в условия производственной деятельности работников умственного труда. Их труд стал более интенсивным, напряженным, требующим значительных затрат умственной, эмоциональной и физической энергии. Это потребовало комплексного решения проблем эргономики, гигиены и организации труда, регламентации режимов труда и отдыха.

Охрана здоровья трудящихся, обеспечение безопасности условий труда, ликвидация профессиональных заболеваний и производственного травматизма составляет одну из главных забот человеческого общества. Обращается внимание на необходимость широкого применения прогрессивных форм научной организации труда, сведения к минимуму ручного, малоквалифицированного труда, создания обстановки, исключающей профессиональные заболевания и производственный травматизм.

Данный раздел дипломного проекта посвящен рассмотрению следующих вопросов:

* организация рабочего места программиста;
* определение оптимальных условий труда программиста.

Описание рабочего места программиста.

Рабочее место - это часть пространства, в котором инженер осуществляет трудовую деятельность, и проводит большую часть рабочего времени. Рабочее место, хорошо приспособленное к трудовой деятельности инженера, правильно и целесообразно организованное, в отношении пространства, формы, размера обеспечивает ему удобное положение при работе и высокую производительность труда при наименьшем физическом и психическом напряжении.

При правильной организации рабочего места производительность труда инженера возрастает с восьми до двадцати процентов.

Согласно ГОСТ 12.2.032-78 конструкция рабочего места и взаимное расположение всех его элементов должно соответствовать антропометрическим, физическим и психологическим требованиям. Большое значение имеет также характер работы. В частности, при организации рабочего места программиста должны быть соблюдены следующие основные условия:

* оптимальное размещение оборудования, входящего в состав рабочего места;
* достаточное рабочее пространство, позволяющее осуществлять все необходимые движения и перемещения;
* необходимо естественное и искусственное освещение для выполнения поставленных задач;
* уровень акустического шума не должен превышать допустимого значения.

Главными элементами рабочего места программиста являются письменный стол и кресло. Основным рабочим положением является положение сидя. Рабочее место для выполнения работ в положении сидя организуется в соответствии с ГОСТ 12.2.032-78.

Рабочая поза сидя вызывает минимальное утомление программиста. Рациональная планировка рабочего места предусматривает четкий порядок и постоянство размещения предметов, средств труда и документации. То, что требуется для выполнения работ чаще, расположено в зоне легкой досягаемости рабочего пространства.

Моторное поле - пространство рабочего места, в котором могут осуществляться двигательные действия человека.

Максимальная зона досягаемости рук - это часть моторного поля рабочего места, ограниченного дугами, описываемыми максимально вытянутыми руками при движении их в плечевом суставе.

Оптимальная зона - часть моторного поля рабочего места, ограниченного дугами, описываемыми предплечьями при движении в локтевых суставах с опорой в точке локтя и с относительно неподвижным плечом*.* На рисунке 6.1 изображено моторное поле, разделенное на соответствующие зоны.

Рассмотрим оптимальное размещение предметов труда и документации в зонах досягаемости рук:

* дисплей размещается в зоне а);
* клавиатура - в зоне г) или д);
* системный блок размещается в зоне б);
* принтер находится в зоне а);
* документация: в зоне легкой досягаемости ладони - в (слева) - литература и документация, необходимая при работе; в выдвижных ящиках стола - литература, неиспользуемая постоянно.

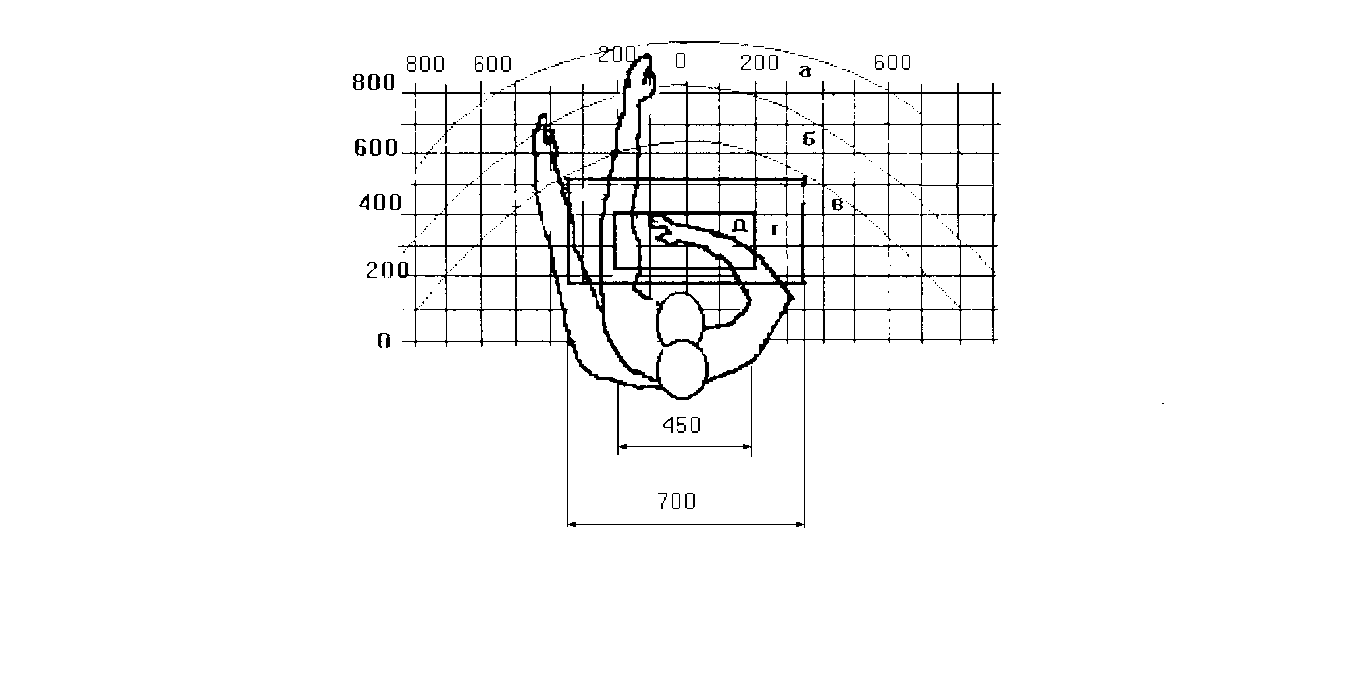
**

Рисунок 6.1 - Зоны досягаемости рук в горизонтальной плоскости

При проектировании письменного стола следует учитывать следующее:

* высота стола должна быть выбрана с учетом возможности сидеть свободно, в удобной позе, при необходимости опираясь на подлокотники;
* нижняя часть стола должна быть сконструирована так, чтобы программист мог удобно сидеть, не был вынужден поджимать ноги;
* поверхность стола должна обладать свойствами, исключающими появление бликов в поле зрения программиста;
* конструкция стола должна предусматривать наличие выдвижных ящиков (не менее трех для хранения документации, листингов, канцелярских принадлежностей, личных вещей).

Параметры рабочего места выбираются в соответствии с антропометрическими характеристиками. При использовании этих данных в расчетах следует исходить из максимальных антропометрических характеристик.

При работе в положении сидя рекомендуются следующие параметры рабочего пространства:

* ширина не менее 700 мм;
* глубина не менее 400 мм;
* высота рабочей поверхности стола над полом 700-750 мм.

Оптимальными размерами стола являются:

* высота 710 мм;
* длина 1300 мм;
* ширина 650 мм.

Поверхность для письма должна иметь не менее 40 мм в глубину и не менее 600 мм в ширину.

Под рабочей поверхностью должно быть предусмотрено пространство для ног:

* высота не менее 600 мм;
* ширина не менее 500 мм;
* глубина не менее 400 мм.

Важным элементом рабочего места программиста является кресло. Оно выполняется в соответствии с ГОСТ 21.889-76. При проектировании кресла исходят из того, что при любом рабочем положении программиста его поза должна быть физиологически правильно обоснованной, т.е. положение частей тела должно быть оптимальным. Для удовлетворения требований физиологии, вытекающих из анализа положения тела человека в положении сидя, конструкция рабочего сидения должна удовлетворять следующим основным требованиям:

* допускать возможность изменения положения тела, т.е. обеспечивать свободное перемещение корпуса и конечностей тела друг относительно друга;
* допускать регулирование высоты в зависимости от роста работающего человека (в пределах от 400 до 550 мм);
* иметь слегка вогнутую поверхность,
* иметь небольшой наклон назад.

Исходя из вышесказанного, приведем параметры стола программиста:

* высота стола 710 мм;
* длина стола 1300 мм;
* ширина стола 650 мм;
* глубина стола 400 мм.

Поверхность для письма:

* в глубину 40 мм;
* в ширину 600 мм.

Важным моментом является также рациональное размещение на рабочем месте документации, канцелярских принадлежностей, что должно обеспечить работающему удобную рабочую позу, наиболее экономичные движения и минимальные траектории перемещения работающего и предмета труда на данном рабочем месте.

Окраска помещений и мебели должна способствовать созданию благоприятных условий для зрительного восприятия, хорошего настроения. В служебных помещениях, в которых выполняется однообразная умственная работа, требующая значительного нервного напряжения и большого сосредоточения, окраска должна быть спокойных тонов - малонасыщенные оттенки холодного зеленого или голубого цветов

При разработке оптимальных условий труда программиста необходимо учитывать освещенность, шум и микроклимат.

Рациональное освещение рабочего места является одним из важнейших факторов, влияющих на эффективность трудовой деятельности человека, предупреждающих травматизм и профессиональные заболевания. Правильно организованное освещение создает благоприятные условия труда, повышает работоспособность и производительность труда. Освещение на рабочем месте программиста должно быть таким, чтобы работник мог без напряжения зрения выполнять свою работу. Утомляемость органов зрения зависит от ряда причин:

* недостаточность освещенности;
* чрезмерная освещенность;
* неправильное направление света.

Недостаточность освещения приводит к напряжению зрения, ослабляет внимание, приводит к наступлению преждевременной утомленности. Чрезмерно яркое освещение вызывает ослепление, раздражение и резь в глазах. Неправильное направление света на рабочем месте может создавать резкие тени, блики, дезориентировать работающего. Все эти причины могут привести к несчастному случаю или профзаболеваниям.

Искусственное освещение выполняется посредством электрических источников света двух видов: ламп накаливания и люминесцентных ламп. Люминесцентные лампы, по сравнению с лампами накаливания имеют существенные преимущества:

* по спектральному составу света они близки к дневному, естественному освещению;
* обладают более высоким КПД;
* обладают повышенной светоотдачей;
* более длительный срок службы.

Параметры микроклимата на рабочем месте

Параметры микроклимата могут меняться в широких пределах, в то время как необходимым условием жизнедеятельности человека является поддержание постоянства температуры тела благодаря свойству терморегуляции, т.е. способности организма регулировать отдачу тепла в окружающую среду.

Основной принцип нормирования микроклимата - создание оптимальных условий для теплообмена тела человека с окружающей средой. В санитарных нормах СН‑245/71 установлены величины параметров микроклимата, создающие комфортные условия. Эти нормы устанавливаются в зависимости от времени года, характера трудового процесса и характера производственного помещения (значительные или незначительные тепловыделения).

В настоящее время для обеспечения комфортных условий используются как организационные методы, так и технические средства. К числу организационных относятся рациональная организация проведения работ в зависимости от времени года и суток, а также организация правильного чередования труда и отдыха. В связи с этим рекомендуется на территории предприятия организовывать зеленую зону со скамейками для отдыха и водоемом (бассейны, фонтаны). Технические средства включают вентиляцию, кондиционирование воздуха, отопительную систему.

Для рабочих помещений с избыточным тепловыделением до 20 ккал/м3 допустимые и оптимальные значения параметров микроклимата приведены в таблице 6.1.

Таблица 6.1 – Допустимые и оптимальные значения параметров микроклимата

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Время года | Зона | Температура  воздуха, 0 C | Относительная  влажность, % | Скорость  движения  воздуха, м/с |
| Холодный  период | Оптимальная | 18 - 21 | 60 - 40 | < 0.2 |
| Переходный  период | Допустимая | 17 - 21 | < 75 | < 0.3 |
| Теплый период года  (t > 100 C) | Оптимальная | 20 - 25 | 60 - 40 | < 0.3 |
| Допустимая | < 28 в 13  часов самого  жаркого мес. | < 75 | < 0.5 |

Нормирование шума.

Установлено, что шум ухудшает условия труда, оказывая вредное воздействие на организм человека. При длительном воздействии шума на человека происходят нежелательные явления: снижается острота зрения, слуха, повышается кровяное давление, понижается внимание. Сильный продолжительный шум может стать причиной функциональных изменений сердечно-сосудистой и нервной систем.

Согласно ГОСТ 12.1.003-88 («Шум. Общие требования безопасности») характеристикой постоянного шума на рабочих местах являются среднеквадратичные уровни давлений в октавных полосах частот со среднегеометрическими стандартными частотами: 63, 125, 250, 500, 1000, 2000, 4000 и 8000 Гц. В этом ГОСТе указаны значения предельно допустимых уровней шума на рабочих местах предприятий. Для помещении конструкторских бюро, расчетчиков и программистов уровни шума не должны превышать соответственно: 71, 61, 54, 49, 45, 42, 40, 38 дБ. Эта совокупность восьми нормативных уровней звукового давления называется предельным спектром.

Методы защиты от шума.

Строительно-акустические методы защиты от шума предусмотрены строительными нормами и правилами (СНиП-II-12-77). это:

* звукоизоляция ограждающих конструкции, уплотнение по периметру притворов окон и дверей;
* звукопоглощающие конструкции и экраны;
* глушители шума, звукопоглощающие облицовки.

На рабочем месте программиста источниками шума, как правило, являются технические средства, как то - компьютер, принтер, вентиляционное оборудование, а также внешний шум. Они издают довольно незначительный шум, поэтому в помещении достаточно использовать звукопоглощение. Уменьшение шума, проникающего в помещение извне, достигается уплотнением по периметру притворов окон и дверей. Под звукопоглощением понимают свойство акустически обработанных поверхностей уменьшать интенсивность отраженных ими волн за счет преобразования звуковой энергии в тепловую. Звукопоглощение является достаточно эффективным мероприятием по уменьшению шума. Наиболее выраженными звукопоглощающими свойствами обладают волокнисто-пористые материалы: фибролитовые плиты, стекловолокно, минеральная вата, полиуретановый поропласт, пористый поливинилхлорид и др. К звукопоглощающим материалам относятся лишь те, коэффициент звукопоглощения которых не ниже 0.2.

Звукопоглощающие облицовки из указанных материалов (например, маты из супертонкого стекловолокна с оболочкой из стеклоткани нужно разместить на потолке и верхних частях стен). Максимальное звукопоглощение будет достигнуто при облицовке не менее 60% общей площади ограждающих поверхностей помещения.

Системы отопления и системы кондиционирования следует устанавливать так, чтобы ни теплый, ни холодный воздух не направлялся на людей. На производстве рекомендуется создавать динамический климат с определенными перепадами показателей. Температура воздуха у поверхности пола и на уровне головы не должна отличаться более, чем на 5 градусов. В производственных помещениях помимо естественной вентиляции предусматривают приточно-вытяжную вентиляцию. Основным параметром, определяющим характеристики вентиляционной системы, является кратность обмена, т.е. сколько раз в час сменится воздух в помещении.

# 7 РЕСУРСО- И ЭНЕРГОСБЕРЕЖЕНИЕ

Ресурсосбережение — это снижение материалоемкости единицы продукции, увеличение выхода конечной продукции, сокращение потерь в производственном процессе путем применения достижений новейшей техники и технологии.

В Республике Беларусь действует Межгосударственный стандарт разработанный Межгосударственным Техническим комитетом по стандартизации МТК 111.

Настоящий стандарт является основополагающим и устанавливает цель, задачи, объекты, основные принципы, термины и классификацию групп требований рационального использования и экономного расходования материальных ресурсов на всех стадиях жизненного цикла веществ, материалов, изделий, продукции при проведении работ и оказании услуг юридическим и физическим лицам.

Целью стандартизации в области ресурсосбережения является создание организационно-методической и нормативной основы, необходимой и достаточной для проведения государственной технической политики, направленной на снижение ресурсоемкости получаемого дохода без ухудшения условий экономического развития страны при безусловном обеспечении высоких потребительских свойств продукции.

Требования ресурсосбережения подразделяют на три группы:

* требования ресурсосодержания, определяющие совершенство процессов, продукции, работ и услуг, например по составу и количеству использованных материалов, массе, габаритам, объему изделия;
* требования ресурсоемкости (по технологичности), определяющие возможность достижения оптимальных затрат ресурсов при изготовлении, ремонте и утилизации продукции, а также выполнении различных работ и оказании услуг с учетом требований экологической безопасности;
* требования ресурсоэкономичности изделия, определяющие возможность достижения оптимальных затрат ресурсов при эксплуатации, ремонте и утилизации продукции, а также при выполнении работ и оказании услуг.

Указанные группы требований взаимосвязаны при:

* разработке продукции, планировании работ и услуг (устанавливают проектные требования ресурсосодержания и ресурсоэкономичности, рекомендации по ресурсоемкости);
* изготовлении продукции, выполнении работ и оказании услуг (устанавливают уточненные (контрольные) требования ресурсоемкости (по технологичности));
* эксплуатации продукции и выполнении работ и оказании услуг (устанавливают уточненные (контрольные) требования ресурсоэкономичности и ресурсоемкости));
* утилизации продукции (устанавливают требования ресурсоемкости и ресурсоэкономичности).

В процессе хозяйственной деятельности ресурсы предприятия занимают одно из центральных мест, поэтому вопросы ресурсосбережения и определения оптимального соотношения ресурсов на предприятии очень актуален в настоящее время. Финансовая политика в области ресурсов направлено воздействует на долговременное состояние предприятия, а так же определяет его текущее состояние. Она диктует тенденции экономического развития, перспективный уровень научно-технического прогресса, состояние производственных мощностей предприятия.

Актуальность данной темы помимо прочего заключается в том, что в процессе хозяйственной деятельности практически все белорусские предприятия сталкиваются с проблемой нехватки ресурсов для обеспечения нормальной работы.

Производство различных благ и вся хозяйственная деятельность базируются на использовании различных экономических ресурсов. Под экономическими ресурсами понимают все виды ресурсов, используемые в процессе товаров и услуг. К ресурсам предприятия относятся:

* земля (природные ресурсы) — капитал предприятия;
* кадровый потенциал;
* предпринимательские способности.

Земля — во-первых, это вообще всякое место, где находится человек: живет, трудится, отдыхает, развлекается и т.п. Во-вторых, на земле как на территории также расположены производственные и другие предприятия. В-третьих, земля, имеющая биологические свойства плодородия, служит объектом сельского и лесного хозяйства. В-четвертых, она является также источником полезных ископаемых, водных и других ресурсов. Говоря о земле как о факторе производства, экономическая теория учитывает все функции природных факторов в хозяйстве.

Основные фонды — это часть производственных фондов, которая вещественно воплощена в средствах труда, сохраняет в течение длительного времени свою натуральную форму, переносит по частям стоимость продукции и возмещается только после проведения нескольких производственных циклов.

В зависимости от назначения основные фонды делятся на:

* основные производственные фонды;
* основные непроизводственные фонды.

К основным производственным относятся фонды, которые непосредственно участвуют в производственном процессе или создают условия для производственного процесса (производственные здания, трубопроводы и др.)

Основные непроизводственные фонды — это объекты бытового и культурного назначения, медицинские учреждения и др.

Оборотные средства — это совокупность денежных средств, авансируемых для создания оборотных производственных фондов и фондов обращения, обеспечивающих непрерывный кругооборот денежных средств.

Далее следует отметить, что к оборотным производственным фондам относятся предметы труда (сырье, основные материалы и полуфабрикаты, вспомогательные материалы, топливо, тара, запасные части, средства труда со сроком службы не более 1 года или стоимостью не более пятидесятикратного установленного минимального размера оплаты труда в месяц (МБП и инструменты), незавершенное производство и расходы будущих периодов.

К фондам обращения относятся средства предприятия, вложенные в запасы готовой продукции, товары отгруженные, но неоплаченные, а также средства в расчетах и денежные средства в кассе и на счетах.

Оборотные производственные фонды вступают в производство в своей натуральной форме и в процессе изготовления продукции целиком потребляются. Они переносят свою стоимость на создаваемый продукт.

Оборотные средства обеспечивают непрерывность производства и реализации продукции.

Фонды обращения связанные с обслуживанием процесса обращения товаров. Они не участвуют в образовании стоимости, а являются ее носителями. После изготовления продукции и ее реализации стоимость оборотных средств возмещается в составе выручки от реализации продукции, что создает возможность систематического возобновления процесса производства. Он осуществляется путем непрерывного кругооборота средств предприятия.

В своем движении оборотные средства проходят последовательно три стадии: денежную, производственную и товарную.

Эффективное использование ресурсов во многом зависит от принципов организации производства. Так ритмичность, слаженность и высокая результативность зависит от оптимальных размеров оборотных средств. Поэтому большое значение приобретает процесс нормирования оборотных средств, который относится к текущему финансовому планированию на предприятии. Для формирования оборотных средств предприятие использует собственные и приравненные к ним средства, а так же привлеченные и заемные пассивы. Источниками формирования оборотных средств могут быть: прибыль, кредиты, акционерный (уставный) капитал, паевые взносы, бюджетные средства, перераспределенные ресурсы, кредиторская задолженность и др. Отдельно выделяется категория денежного капитала.

Финансовые ресурсы — это денежные средства, имеющиеся в распоряжении предприятия и предназначенные для осуществления текущих затрат по расширенному воспроизводству для выполнения финансовых обязательств и экономического стимулирования работающих. Финансовые ресурсы направляются так же на содержание и развитие объектов непроизводственной сферы, потребление, накопление в специальные резервные фонды и др.

Формирование финансовых ресурсов происходит за счет целого ряда источников. Первоначальное формирование финансовых ресурсов происходит в момент учреждения предприятия, когда образуется уставный капитал. В основном же финансовые ресурсы формируются за счет прибыли, а также перечисленных в выше изложенной схеме источников. Кадры или трудовые ресурсы предприятия — совокупность работников различных профессионально-квалификационных групп, занятых на предприятии и входящих в его списочный состав. Трудовые ресурсы приводят в движение материальные элементы производства, создают продукт, стоимость и прибавочный продукт в форме прибыли.

Отличие данного вида ресурсов от других заключается в том, что каждый наемный работник может отказаться от предложенных условий и потребовать изменения условий труда, переобучения другим профессиям, может уволиться с предприятия по собственному желанию. Кадровый состав предприятия и его изменения имеют определенные количественные, качественные и структурные характеристики, которые могут быть с меньшей или большей степенью достоверности изменены и отражены следующими абсолютными и относительными показателями:

* списочная и явочная численность работников предприятия и его внутренних подразделений отдельных категорий и групп на определенную дату;
* среднесписочная численность работников предприятия и его внутренних подразделений за определенный период;
* удельный вес работников отдельных подразделений в общей численности работников предприятия;
* темпы роста (прироста) численности работников предприятия за определенный период;
* средний разряд рабочих предприятия;
* удельный вес служащих, имеющих высшее или среднее специальное образование в общей численности служащих и работников предприятия;
* средний стаж работы по специальности руководителей и специалистов предприятия;
* текучесть кадров по приему и увольнению работников;
* фондовооруженность труда работников и рабочих на предприятии и др.

Совокупность перечисленных и ряда других показателей может дать представление о количественном, качественном и структурном состоянии персонала предприятия и тенденциях их изменения для целей управления персоналом, в том числе планирования, анализа и разработки мероприятий по повышению эффективности использования трудовых ресурсов предприятия.

Эффективность использования трудовых ресурсов предприятия характеризует производительность труда, которая определяется количеством продукции, произведенной в единицу рабочего времени, или затратами труда на единицу произведенной продукции или выполненной работы.

В современной рыночной экономике и жесткой конкуренции, в условиях переходного периода, довольно актуальным стал вопрос об экономии и рациональном использовании ресурсов.

В последнее десятилетие проблема экономии ресурсов на предприятии особенно обострилась. Необходимо осуществлять техническое перевооружение или реконструкцию действующих предприятий — перевести их на ресурсосберегающие технологии.

Ресурсосбережение — это совокупность мер по экономному и эффективному использованию всех факторов производства, общее свойство которых состоит в потенциальной возможности их участия в производстве (производственные ресурсы) и в потреблении (потребительские ресурсы). Ресурсосбережение означает использование всех видов ресурсов (материальных, трудовых, природных, финансовых и других) для решения задач экономического и социального развития. Поскольку потребности людей и общества стремительно растут, а ресурсы ограничены и редки, то роль ресурсосбережения в решении коренной триединой проблемы: что, как, для кого производить все возрастает. Ресурсосбережение охватывает не только факторы производства, но и продукцию, поскольку продукция одной отрасли потребляется в другой, связанной с ней общественным разделением труда.

Ресурсосбережение предусматривает удовлетворение потребности народного хозяйства в их приросте преимущественно за счет экономии. Достигается это путем комплексного использования ресурсов, устранения потерь при добыче, транспортировке и хранении, сокращении отходов при переработке, более широкого вовлечения в хозяйственный оборот вторичных ресурсов и попутных продуктов, путем улавливания ценных продуктов из отходящих газов и водных стоков, утилизации отбросов и др. Оно должно обеспечиваться на всех стадиях производства: при добыче, транспортировке, хранении, погрузке- разгрузке, разделке, переработке и т.п.

Соблюдение ресурсосбережения — важная характеристика качества техники и технологии. Техника считается ресурсосберегающей, если она требует меньше расхода ресурсов на изготовление и эксплуатацию. Ресурсосберегающей технологией называют технологию малоотходную или безотходную. Необходимость ресурсосбережения вызвана дефицитом многих видов ресурсов, истощением их запасов в природе, значительным удорожанием добычи и другими факторами.

В связи с переходом к интенсивному ресурсосберегающему типу экономического роста, основанного на использовании достижений НТР, снижении фондоемкости и материалоемкости продукции, повышения производительности труда, улучшении техникоэкономических показателей и качества продукции возрастают возможности ресурсосбережения. Важное значение в решении проблемы ресурсосбережения имеет научно технический прогресс.

В целях укрепления экономической безопасности государства 14 июня 2007 года Президентом Республики Беларусь подписана Директива №3 «Экономия и бережливость — главные факторы экономической безопасности государства».

Государственное регулирование в области энергосбережения и повышения энергетической эффективности осуществляется путем установления:

* требований к обороту отдельных товаров, функциональное назначение которых предполагает использование энергетических ресурсов;
* запретов или ограничений производства и оборота в Республике Беларусь товаров, имеющих низкую энергетическую эффективность, при условии наличия в обороте или введения в оборот аналогичных по цели использования товаров, имеющих высокую энергетическую эффективность, в количестве, удовлетворяющем спрос потребителей;
* обязанности по учету используемых энергетических ресурсов;
* требований энергетической эффективности зданий, строений, сооружений;
* обязанности проведения обязательного энергетического обследования;
* требований к энергетическому паспорту;
* обязанности проведения мероприятий по энергосбережению и повышению энергетической эффективности в отношении общего имущества собственников помещений в многоквартирном доме;
* требований энергетической эффективности товаров, работ, услуг, размещение заказов на которые осуществляется для государственных или муниципальных нужд;
* требований к региональным, муниципальным программам в области энергосбережения и повышения энергетической эффективности;
* требований к программам в области энергосбережения и повышения энергетической эффективности организаций с участием государства или городского образования и организаций, осуществляющих регулируемые виды деятельности;
* основ функционирования государственной информационной системы в области энергосбережения и повышения энергетической эффективности;
* обязанности распространения информации в области энергосбережения и повышения энергетической эффективности;
* обязанности реализации информационных программ и образовательных программ в области энергосбережения и повышения энергетической эффективности.

Энергосбережение (экономия электроэнергии) — реализация правовых, организационных, научных, производственных, технических и экономических мер, направленных на эффективное (рациональное) использование (и экономное расходование) топливно-энергетических ресурсов и на вовлечение в хозяйственный оборот возобновляемых источников энергии Энергосбережение. Энергосбережение — важная задача по сохранению природных ресурсов.

Эффекты от мероприятий энергосбережения можно разделить на несколько групп:

* экономические эффекты у потребителей (снижение стоимости приобретаемых энергоресурсов);
* эффекты повышения конкурентоспособности (снижение потребления энергоресурсов на единицу производимой продукции, энергоэффективность производимой продукции при ее использовании);
* эффекты для электрической, тепловой, газовой сети (снижение пиковых нагрузок, минимизация инвестиций в расширение сети);
* экологические эффекты;
* связанные эффекты (внимание к проблемам энергосбережения приводит к повышению озабоченности проблемами общей эффективности системы — технологии, организации, логистики на производстве, системы взаимоотношений, платежей и ответственности в ЖКХ, отношения к домашнему бюджету у граждан).

Повышение энергоемкости производства, количества техники, задействованной в производственных процессах, а также постоянный рост цен на энергоносители является серьезным фактором, увеличивающим важность вопроса об экономии электроэнергии.

Универсальных способов экономить электроэнергию на данный момент не существует, но разработаны методики, технологии и устройства, помогающие вывести энергосбережение на качественно новый уровень.

Вопрос экономии электроэнергии многоплановый и нужен стратегический подход, для того чтобы максимально эффективно использовать все производственные мощности при минимально возможных энергетических затратах. Подход к экономии электроэнергии основан на использовании энергосберегающих технологий, которые призваны уменьшить потери электроэнергии. Существует немало устройств, которые позволяют добиться уменьшения потерь при работе оборудования, основными из которых являются конденсаторные установки и частотно-регулируемые приводы, при эксплуатации различных бытовых осветительных приборов и устройств охранной сигнализации, приборов таймерного типа, позволяющих автоматически отключать различные электроустановки при перерывах в работе и быстро их включать в нужный момент времени.

Особо актуально использование энергосберегающего оборудования на предприятиях тяжелой промышленности и на крупных производственных комплексах, где нерациональное потребление электроэнергии ведет к огромным финансовым потерям. Также разумным является использование энергосберегающих технологий в плане повышения качества электроэнергии, что положительно отражается на качестве работы оборудования, на сроке его службы.

ЗАКЛЮЧЕНИЕ

В процессе выполнения дипломного проекта был разработан программный продукт, предоставляющий пользователю возможность ознакомится с продукцией выпускаемой ЗАО «Инволюкс». Предоставляется возможность ознакомится с ассортиментом рядом продукции, а также получить визуальное представление о располажении и компоновке составных частей мебели.

Удобный, интуитивно-понятный интерфейс позволяет пользователю легко и быстро найти нужную информацию. Асинхронная загрузка и кеширование изображения позволяют экономить время и ресурсы девайса. Эффект перелистывания изображения значительно улучшает внешний вид и не требует никаких лишних действий, что положительно сказывается на настроении пользователя. Динамически масштабируемые элементы делают приложение универсальным по отношению ко всем размерам и разрешениям экрана.

Программный продукт разработан на основе программной платформы Java, что делает его платформенно-независимым.

Для разработанной программы был проведен расчет экономического эффекта от внедрения системы на основе ставок налогов, расценок по оплате труда, действующих в республике Беларусь по состоянию на сегодняшний день. И рассчитана отпускная цена спроектированного программного продукта.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Голощапов А.Л., Google Android. Программирование для мобильных устройств / А.Л. Голощапов - Санкт-Петербург: БХВ, 2012 год - 448 с.
2. Майер Р., Программирование приложений для планшетных компьютеров и смартфонов Android / Р. Майе – Москва: Эксмо, 2011год - 671 с.
3. Голощапов А.Л., Google Android. Создание приложений для смартфонов и планшетных ПК / А.Л. Голощапов - Санкт-Петербург: БХВ, 2013 год - 832 с.
4. Варакин М.В., Разработка мобильных приложений под Android / М.В. Варакин – Москва: Специалист, 2012 год - 128 с.
5. Библиотека Picasso // Веб-сервис для хостинга IT-проектов и их совместной разработки [Электронный ресурс]. – 2016. – Режим доступа: <https://github.com/square/picasso>. – Дата доступа: 12.04.2016.
6. Компонент ViewPager // Официальный сайт разработчиков Android [Электронный ресурс]. – 2008. – Режи доступа: <http://android-developers.blogspot.com.by/2011/08/horizontal-view-swiping-with-viewpager.html> - Дата доступа: 20.04.2016.
7. Боковое меню // Веб-сервис для хостинга IT-проектов и их совместной разработки [Электронный ресурс]. – 2016. – Режим доступа: <https://github.com/rudsonlive/NavigationDrawer-MaterialDesign>. – Дата доступа: 12.04.2016.
8. ГОСТ 19.301-79. Программа и методика испытаний. Требования к содержанию и оформлению.
9. ГОСТ 19.502-78. Описание применения. Требования к содержанию и оформлению.
10. ГОСТ 2.105-95 ЕСПД. Общие требования к текстовым документам.