Подп. и дата

Инв. № дубл.

Взаим. инв. №

Подп. и дата

Инв. № подл.

Министерство образования Республики Беларусь

Учреждение образования

«Минский государственный высший

радиотехнический колледж»

**Обучающе-тестирующее средство для проверки знаний,**

**умений и навыков учащихся младших курсов ССузов в**

**области белорусской литературы**

ОТчет

по производственной технологической практике

|  |  |
| --- | --- |
| Руководители | / А. А. Орешко /  / С.П. Зиньков / |
|  |  |
| Разработчик | / А. Н. Невядомская / |

Минск 2012

Содержание

Изм.

Лист

№ докум.

Подпись

Дата

Лист

2

Отчет по технологической практике

Разраб.

Невядомская

Пров.

Орешко

Пров.

Зиньков

Н.контр.

Утв.

*Обучающе-тестирующее средство для проверки знаний,*

*умений и навыков учащихся младших курсов ССузов в*

*области белорусской литературы*

Пояснительная записка

Лит.

Листов

32

*МГВРК*

У

[Введение 3](#_Toc318404095)

[1 Гомельский радиозавод. Основные виды продукции 4](#_Toc318404096)

[2 Описание структуры АСУП 7](#_Toc318404097)

[3 Должностные обязанности оператора ЭВМ, техника-программиста 13](#_Toc318404098)

[3.1 Должностные обязанности оператора ЭВМ 13](#_Toc318404099)

[3.2 Должностные обязанности техника-программиста 14](#_Toc318404100)

[4 Техно-рабочий проект 17](#_Toc318404101)

[4.1 Постановка задачи 17](#_Toc318404102)

[4.2 Выбор языка программирования 17](#_Toc318404103)

[4.3 Проектирование 20](#_Toc318404104)

[4.4 Алгоритм решения задачи 21](#_Toc318404105)

[4.5 Руководство программиста 23](#_Toc318404106)

[Заключение 28](#_Toc318404107)

[Литература 29](#_Toc318404108)

[Приложение А 30](#_Toc318404109)

[Приложение Б 31](#_Toc318404110)

[Приложение В 32](#_Toc318404111)

# Введение

Производственная технологическая практика состоит из двух этапов:

* изучения технологического процесса обработки информации в подразделениях ВЦ или подразделениях предприятия;
* работы на штатных рабочих местах в соответствии с рабочей профессией,

Для прохождения производственной технологической практики я была направлена в ОАО «Гомельский радиозавод».

Гомельский радиозавод – современное технически оснащенное предприятие. Основано в 1969 году. Уникальность создаваемых систем позволила накопить богатый научно-технический потенциал и развить широкий спектр технологических возможностей в области:

* изготовления высокочастотных и наборных кабелей;
* изготовления волноводных трактов и узлов круглого и прямоугольного сечения;
* освоения основных видов сварки и пайки металлов и цветных сплавов;
* гальванопластики;
* изготовления гальвано- и лакокрасочных покрытий, в том числе порошковых;
* освоения основных видов литья пластмасс и металлов;
* создания печатного и объёмного радиомононтажа;
* создания механообрабатывающего производства (с возможностью обработки до 4000 мм по диаметру и прямоугольного сечения размером 8000х2500 мм и весом до 10 тонн);
* создания холодноштамповочного производства (с усилием прессования до 400 тонн и габаритами штампуемых деталей до 1000 мм).
* Цели практики:
* закрепление знаний, связанных с технологией обработки информации;
* изучение состава технических средств и математического обеспечения ЭВМ данного ВЦ;
* закрепление и углубление знаний, полученных при изучении специальных предметов;
* закрепление практических навыков по составлению алгоритмов и реализации их на ЭВМ;
* приобретение навыков оперативного управления технологическим процессом обработки информации.

# 1 Гомельский радиозавод. Основные виды продукции

Гомельский радиозавод создан по приказу Минрадиопрома СССР № 381 от 3 октября 1966 г. В соответствии с правительственным соглашением № 3 от 28 января 1992 между Россией, Украиной и Постановлением Правительства Республики Беларусь № 6 от 5 января 1993 года предприятие является членом Международной акционерной корпорации «Вымпел». Имущество предприятия является государственной собственностью Республики Беларусь и предоставляется в распоряжение корпорации.

Предприятие имеет 7 направлений продукции:

* + мясоперерабатывающие комплексы;
  + холодильные установки;
  + телевизионная техника;
  + медицинская техника;
  + аппаратура связи;
  + товары народного потребления;
  + промышленное оборудование.

Технологические возможности предприятия очень широки:

* + печатный и объёмный радиомонтаж;
  + производство кабелей и разъёмов к ним;
  + вязка жгутов для объёмного монтажа;
  + производство волноводных узлов круглого и прямоугольного сечения, метрового, дециметрового и миллиметрового диапазона;
  + сварка чёрных, цветных металлов и нержавеющей стали в среде защитных газов ( ручная и полуавтоматическая), пайка волноводов;
  + производство трубопроводов, устойчивых к агрессивной среде из облучённого полиэтилена;
  + производство волноводных узлов методом гальванопластики;
  + гальвано- и лакокрасочное покрытие:
    - вакуумная металлизация алюминиевого литья и пластмассовых деталей;
    - меднение деталей из стали, меди и медных сплавов;
    - серебрение деталей из стали, меди и медных сплавов;
    - хромирование деталей из стали, меди и медных сплавов;
    - цинкование деталей из стали;
    - кадмирование деталей из стали;
    - нанесение лакокрасочных покрытий «под золото» на пластмассовые детали;
    - нанесение лакопокрытий на узлы печатного и объёмного монтажа;
    - нанесение полимерной порошковой краски;
  + литьё в кокиль алюминия и латуни;
  + литьё под давлением алюминиевых сплавов (площадь проекции до 1000 мм, масса отливки до 6 кг);
  + литьё стали, чугуна, алюминия и бронзы в песчаные формы;
  + литьё пластмасс: термопластавтоматы с объёмом отливки 63, 95, 125, 323, 500 см3 и поверхностью отливки до 500см2.
  + прессование пластических материалов (термопластов, реакторопластов, резины) с усилием до 400 т;
  + нанесение на детали полимерных материалов.

Также имеются:

* лаборатория климатических испытаний;
* лаборатория проверки надёжности;
* лаборатория химического анализа органического материалов ( смол, полиэфиров, лаков, красок, клеев, флюсов, гальванопокрытий);
* лаборатория по изготовлению
* лаборатория электрофизических, физико-механических, металлографических и спектральных анализов материалов и изделий;
* лаборатория анализа электролитов и растворов процесса гальванопокрытий и анализа растворов на содержание драгоценных металлов.

В механообрабатывающем производстве предприятия имеются токарные станки с диаметром обработки до 1000 мм, карусельные станки с диаметром обработки до 4000 мм, расточные станки для обработки деталей с размерами до 8000 на 2500 мм.

Предприятие может производить сварочно-каркасные конструкции из стали и алюминиевых сплавов габаритными размерами 10000 на 2500 на 2500 мм.

Уникальность создаваемых заводом систем позволила накопить богатый научно-технический потенциал и развить широкий спектр технологических возможностей в области:

- изготовления высокочастотных и наборных кабелей;

- изготовления волноводных трактов и узлов круглого и прямоугольного сечения;

- освоения основных видов сварки и пайки металлов и цветных сплавов;

- гальванопластики;

- изготовления гальвано- и лакокрасочных покрытий, в том числе порошковых;

- освоения основных видов литья пластмасс и металлов;

- создания печатного и объёмного радиомонтажа;

- создания механообрабатывающего производства (с возможностью обработки до 4000 мм по диаметру и прямоугольного сечения размером 8000х2500 мм и весом до 10 тонн);

- создания холодноштамповочного производства (с усилием прессования до 400 тонн и габаритами штампуемых деталей до 1000 мм).

Благодаря процессам конверсии в 1989 году завод перешел на выпуск продукции производственно-технического назначения и товаров народного потребления.

Предприятие производит сельхозтехнику, технику специального назначения, товары народного потребления, выполняет работы по механизированной обработке сварных металлоконструкций.

# 2 Описание структуры АСУП

Автоматизированная система управления предприятием – это совокупность экономико-математических методов, технических средств (ЭВМ, средств связи, устройств отображения информации, передачи данных и т.д.) и организационных комплексов, обеспечивающих рациональное управление предприятием.

Наиболее важная цель построения всякой АСУ – резкое повышение эффективности управления объектом (производственным, административным и т.д.) на основе роста производительности управленческого труда и совершенствования методов планирования и гибкого регулирования управляемого процесса.

АСУП состоит из основы и функциональной части. Обобщённая структурная схема АСУП представлена на рисунке 1.

**Основа АСУП**

**Техническая база**

Средства отображения информации

Средства сбора и регистрации данных

ЭВМ

ЭВМ

ЭВМ

**Организационно-экономическая база**

Организационно-экономическая концепция

Организация производства и управления

Организационно-экономическая модель

Организационно-правовое обеспечение

**Информационная база**

Генеральный массив

Производный массив

Оперативный массив

**Математическое обеспечение**

Система программирования

Операционная система

Общесистемный комплекс

Пакеты типовых комплексов

**Функциональная часть**

Подсистема технической подготовки

производства

Подсистема управления

качеством

Подсистема технико-экономического

планирования

Подсистема оперативно-производственного

планирования, учета и регулирования

Подсистема материально-технического

обеспечения

Подсистема финансово-бухгалтерской

деятельности

Подсистема сбыта

продукции

Подсистема транспортного

хозяйства

Подсистема

кадров

Подсистема вспомогательных

служб

Рисунок 1 – Структурная схема автоматизированной системы управления предприятием

Основу АСУП составляют информационная база, техническая база, математическое обеспечение, организационно-экономическая база. Основа – общая часть для всех задач, решаемых АСУП.

Информационная база АСУП – размещенная на машинных носителях информации совокупность всех данных, необходимых для автоматизации управления объектом или процессом. Обычно информационная база делится на три массива: генеральный, производный и оперативный. Конструкция массивов и их полей (способы размещения на носителях, особенности взаимосвязи данных внутри массива, конкретная компоновка данных и т.д.) определяется типом АСУП и общими характеристиками объектов, для которых она предназначается. Однако целесообразно сохранять типовое конструктивное построение информационной базы для общего класса объектов (например, для машиностроительных предприятий). Генеральный массив объединяет данные, являющиеся общими для всех задач, размещение которых отвечает универсальной структуре, не ориентированной на выполнение какой-либо одной функции управления. Генеральный массив для крупного объекта содержит сотни миллионов символов, занимает большие объёмы запоминающих устройств и не всегда удобен для использования в каждой конкретной задаче, требующей для своего решения специализированной информации. Эта проблема осложняется при мультипрограммной обработке данных и недостаточно ёмких оперативных запоминающих устройствах, предполагающих хранение многих массивов в машинных архивах, функционально разобщённых с процессорами. В связи с этим в реально функционирующих АСУП возникает необходимость формирования производных массивов, отражающих специфику структуры объекта, особенности выполняемых в каждый период функций, частоту повторяемости различных задач и ряд др. факторов, связанных с текущей работой системы. Все производные массивы, как правило, формируются из генерального массива. Всякое устойчивое изменение характеристик обслуживаемого объекта должно быть отражено в генеральном массиве. Оперативный массив охватывает текущую информацию, а также промежуточные результаты вычислений. В нём же размещается первичная информация о состоянии обслуживаемого объекта, поступающая периодически по каналам связи или записанная на автономных носителях. Обработанные и обобщённые данные могут затем вноситься в производный и генеральный массивы либо непосредственно выдаваться потребителю.

Техническая база АСУП включает средства обработки, сбора и регистрации, отображения и передачи данных, а также исполнительные механизмы, непосредственно воздействующие на объекты управления (например, автоматические регуляторы, датчики и т.д.), обеспечивающие сбор, хранение и переработку информации, а также выработку регулирующих сигналов во всех контурах автоматизированного управления производством. Основные элементы технической базы – ЭВМ, которые обеспечивают накопление, хранение и обработку данных, циркулирующих в АСУП. ЭВМ позволяют оптимизировать параметры управления, моделировать производство, подготавливать предложения для принятия решения. Обычно выделяют два класса ЭВМ, используемых в АСУП: информационно-расчётные и учётно-регулирующие. Информационно-расчётные ЭВМ находятся на высшем уровне иерархии управления и обеспечивают решение задач, связанных с централизованным управлением объектом по основным планово-экономическим, обеспечивающим и отчётным функциям (технико-экономическое и оперативно-производственное планирование, материально-техническое снабжение, сбыт продукции и т.д.). Они характеризуются высоким быстродействием, наличием системы прерываний, слоговой обработкой данных, и т.д., а также широким набором и большим объёмом запоминающих устройств.

Учётно-регулирующие ЭВМ, как правило, относятся к нижнему уровню управления. Они размещаются в цехах или на участках и обеспечивают сбор информации от объектов управления (станков, складов и т.д.), первичную переработку этой информации, передачу данных в информационно-расчётную ЭВМ и получение от неё директивно-плановой информации, осуществление локальных расчётов (например, расписания работы каждого станка и рабочего, графика подачи комплектующих изделий и материалов, группировки деталей в партии, режимов обработки и т.д.) и выработку управляющих воздействий на объекты управления при отклонении режимов их функционирования от расчётных. Особенность учётно-регулирующих ЭВМ – хорошо развитая система автоматического сопряжения с большим числом источников информации (датчиков, регистраторов) и регулирующих устройств. Их вычислительная часть менее развита, поскольку первично обработанная информация передаётся в ЭВМ верхнего уровня для дальнейшего использования и длительного хранения.

Средства сбора и регистрации данных при участии человека включают различные регистраторы производства, с помощью которых осуществляются сбор и регистрация данных непосредственно на рабочих местах (например, в цехе, на участке, станке), а также датчики (температуры, количества изготовленных деталей, времени работы оборудования и т.д.), фиксаторы нарушений установленного технологического и организационного ритма (отсутствие заготовок, инструмента, материалов, неправильная наладка станков, отсутствие транспортных средств для отправки готовой продукции и т.д.).

Средства отображения информации предназначены для представления результатов обработки информации в удобном для практического использования виде. К ним относятся различные печатающие устройства, которые непосредственно связаны с ЭВМ или с регистраторами производства и выдают либо регулярную (регламентную), либо эпизодическую (по запросу или в случае аварийной ситуации) справочную, директивную или предупредительную информацию.

Аппаратура передачи данных осуществляет обмен информацией между различными элементами АСУП (между регистраторами производства и ЭВМ, между координационно-управляющим центром и цеховыми ЭВМ и т.д.), а также между АСУП и смежными управления уровнями.

К технической базе АСУП относят также средства оргтехники (копировально-множительную технику, картотеки, диктофоны и т.д.), а также вспомогательные и контрольно-измерительные средства, обеспечивающие нормальное функционирование основных технических средств в требуемых режимах.

Математическое обеспечение АСУП – комплекс программ регулярного применения, управляющих работой технических средств и функционированием информационные базы и обеспечивающих взаимодействие человека с техническими средствами АСУП. Математическое обеспечение условно можно подразделить на систему программирования, операционную систему, общесистемный комплекс и пакеты типовых модулей.

Система программирования обеспечивает трансляцию программы решения задачи, выраженной на удобном для человека формализованном языке, на машинный язык, её отладку, редактирование и включение в пакет программ для обработки. В систему программирования входят описания языков программирования, комплекс трансляторов, библиотека стандартных подпрограмм, программы редактирования связей, наборы программ, осуществляющих преемственность (программную) ЭВМ различных типов. Кроме того, система программирования обычно содержит в своём составе набор программ, облегчающих взаимодействие пользователя с машиной и позволяющих системе программирования развиваться в зависимости от характера задач, решаемых потребителем.

Операционные системы обеспечивают функционирование всех устройств ЭВМ в требуемых режимах и выполнение необходимой последовательности заданий на реализацию различных процедур управления. Операционные системы, как правило, являются неотъемлемой составной частью тех вычислительных средств, которые входят в состав АСУП. Однако в ряде случаев при проектировании АСУП приходится расширять операционные системы для обеспечения специальных системных требований (например, при подключении к системе специфичных для управляемого процесса регистраторов и систем отображения, при организации диалоговых режимов между терминалами и центральным вычислительным комплексом). В этой связи очень важной составной частью операционной системы АСУП является т. н. генератор систем. Это – программа, которая не входит в состав активной части управляющих программ и не связана непосредственно с процессом вычислений, но, с помощью которой можно автоматически генерировать комплекс управляющих программ для системы любой конфигурации. Такой метод оказывается особенно эффективным при использовании ЭВМ в широком диапазоне АСУП на различных уровнях и на различных объектах, когда состав ЭВМ и состав решаемых задач может быть существенно различным.

Общесистемный комплекс охватывает набор программ, управляющих работой вычислительной системы и периферийных устройств (регистраторов, средств отображения результатов обработки данных и т.д.). Этот комплекс содержит программы совместной работы нескольких ЭВМ, программы обслуживания каналов связи, дистанционные решения задач в режиме разделения времени, разграничения доступа к информационным массивам и др.

Пакеты типовых прикладных модулей (стандартных подпрограмм) могут использоваться в различных комбинациях при решении той пли иной функциональной задачи. Типовыми, например, являются прикладные модули сортировки данных, статистической обработки информации, обработки сетевых графиков планирования и управления, моделирования реальных процессов и др. К математическому обеспечению АСУП часто относят также программы функционального анализа системы, обеспечивающие удобство эксплуатации и совершенствования системы.

Под организационно-экономической базой понимается совокупность экономических принципов, методов организации производства и управления, схем взаимодействия задач управления на основе правовых документов. Сюда входят организационно-экономический состав и способы формирования технико-экономических показателей управляемого объекта, а также основные принципы повышения эффективности его функционирования и место АСУП в общей системе планирования, учёта и регулирования; организация производства, труда и управления, определяющая рациональную структуру объекта (цеха, отдела и т.д.), порядок реализации технологических маршрутов, наиболее благоприятные условия работы, сохраняющие высокую работоспособность рабочих и служащих, а также научно обоснованную систему управления объектом, чёткие положения обо всех подразделениях, их подчинённости, обязанностях сотрудников и их ответственности; организационно-экономическая модель, предусматривающая построение схемы взаимодействия основных задач АСУП, структуры информационного потока, а также методическое обеспечение порядка реализации задач и использования результатов их решения; организационно-правовое обеспечение (правовые основы и нормы создания и использования АСУП, правовой статус циркулирующей в АСУП информации, а также права и ответственность должностных лиц). Кроме того, организационно-экономическая база включает методические и инструктивные материалы, определяющие влияние АСУП на основные показатели функционирования объекта, оценку эффективности и пути дальнейшего развития АСУП.

Функциональная часть АСУП состоит из набора взаимосвязанных программ для реализации конкретных функций управления (планирование, финансово-бухгалтерскую деятельность и др.). Все задачи функциональной части базируются на общих для данной АСУП информационных массивах и на общих технических средствах. Включение в систему новых задач не влияет на структуру основы и осуществляется посредством типового для АСУП информационного формата и процедурной схемы. Функциональную часть АСУП принято условно делить на подсистемы в соответствии с основными функциями управления объектом. Подсистемы в свою очередь делят на комплексы, содержащие наборы программ для решения конкретных задач управления в соответствии с общей концепцией системы. В АСУП выделяют следующие подсистемы: технической подготовки производства; управления качеством продукции; технико-экономического планирования; оперативно-производственного планирования; материально-технического обеспечения; сбыта продукции; финансово-бухгалтерской деятельности; планирования и расстановки кадров; управления транспортом; управления вспомогательными службами. Деление функциональной части АСУП на подсистемы весьма условно, т.к. процедуры всех подсистем тесно взаимосвязаны и в ряде случаев невозможно провести чёткую границу между различными функциями управления (например, между технико-экономическим планированием, оперативно-производственным планированием и материально-техническим обеспечением). Выделение подсистем используется для удобства распределения работ по созданию системы и для привязки к соответствующим организационным звеньям объекта управления. Структура функциональной части АСУП зависит от схемы процедур управления, определяющей взаимосвязь всех элементов управления и охватывающей автоматизированные, частично механизированные и ручные процедуры. Функциональная часть более мобильна, чем основа, и допускает изменение состава и постановки задач при условии обеспечения стандартного сопряжения с базовыми элементами системы.

# 3 Должностные обязанности оператора ЭВМ, техника-программиста

## 3.1 Должностные обязанности оператора ЭВМ

Оператор ЭВМ относится к категории технических работников.

На должность оператора ЭВМ назначается лицо, имеющее среднее специальное образование и стаж работы не менее двух лет.

Оператор ЭВМ должен знать:

* приказы, указания, распоряжения, инструкции и другие нормативно-распорядительные документы, регламентирующие работу оператора ЭВМ;
* правила эксплуатации ПК и правила обслуживания периферийных устройств компьютера;
* правила оформления документов, в том числе деловой документации с использованием типовых форм;
* правила орфографии и пунктуации;
* правила ведения делопроизводства;
* программное обеспечение (правила работы с Windows, Microsoft Office и т.д.)
* средства вычислительной техники, коммуникаций и связи;
* культуру труда и этику делового общения;
* основы законодательства о труде и охране труда Республики Беларусь;
* устав предприятия, его штатное расписание, правила внутреннего трудового распорядка;
* правила и нормы охраны труда, техники безопасности, производственной санитарии и противопожарной защиты.

Должностные обязанности оператора ЭВМ:

* осуществление технической подготовки документации, необходимой в процессе работы предприятия;
* выполнение работы, связанной с копированием, распечаткой и сканированием документов при помощи периферийных устройств компьютера;
* выполнение набора различных текстов с соблюдением правил орфографии и пунктуации, а также стандартов оформления организационно-распорядительной документации;
* осуществление работы с электронной почтой (получение входящих электронных писем и своевременная отправка исходящих);
* систематизация документов;
* занесение в компьютерные базы данных информации, необходимой для работы завода.
* соблюдение правил личной и производственной гигиены;
* своевременное информирование руководства о необходимости приобретения материалов, непосредственно относящихся к производственному процессу.

Оператор ЭВМ обязан:

* своевременно отвечать на запросы других сотрудников по направлению профессиональной деятельности, предоставлять требуемую информацию в полном объеме;
* соблюдать установленные сроки исполнения заданий и поручений;
* оказывать помощь коллегам по работе при решении задач их деятельности в случае, если помощь может привести к качественному улучшению результатов деятельности;
* добросовестно выполнять возложенные на него обязанности;
* содержать находящееся в пользовании имущество в целости и сохранности;
* хранить служебную и коммерческую тайны;
* соблюдать правила внутреннего трудового распорядка, охраны труда, техники безопасности, производственной санитарии и противопожарной защиты.

## 3.2 Должностные обязанности техника-программиста

Техник-программист относится к категории специалистов.

На должность техника-программиста назначается лицо, имеющее среднее профессиональное образование без предъявления требований к стажу работы;

Назначение на должность техника-программиста и освобождение от нее производится приказом начальника ВЦ (ИВЦ) по представлению (начальника производственного отдела ВЦ (ИВЦ); начальника отдела разработок и внедрения ВЦ (ИВЦ); иного структурного подразделения предприятия).

Техник-программист должен знать:

* методы проектирования механизированной и автоматизированной обработки информации;
* средства вычислительной техники, сбора, передачи и обработки информации и правила их эксплуатации;
* технологию механизированной и автоматизированной обработки информации;
* рабочие программы, инструкции, макеты и другие руководящие материалы, определяющие последовательность и технику выполнения расчетных операций;
* виды технических носителей информации, правила их хранения и эксплуатации;
* действующие системы счислений, шифров и кодов;
* основные формализованные языки программирования;
* основы программирования;
* методы проведения расчетов и вычислительных работ;
* методы расчета выполненных работ;
* основы экономики, организации производства, труда и управления;
* основы трудового законодательства;
* правила внутреннего трудового распорядка;
* правила и нормы охраны труда.

Техник-программист подчиняется непосредственно начальнику производственного отдела ВЦ (ИВЦ).

Должностные обязанности техника-программиста:

* выполнение работы по обеспечению механизированной и автоматизированной обработки поступающей в вычислительный (информационно-вычислительный) центр (далее ВЦ (ИВЦ)) информации, разработки технологии решения экономических и других задач производственного и научно-исследовательского характера;
* активное участие в проектировании систем обработки данных и систем математического обеспечения машины;
* выполнение подготовительных операций, связанных с осуществлением вычислительного процесса, ведение наблюдений за работой машин;
* составление простых схем технологического процесса обработки информации, алгоритмов решения задач, схем коммутации, макетов, рабочих инструкций и необходимых пояснений к ним;
* разработка программ решения простых задач, их отладка и экспериментальная проверка отдельных этапов работ;
* выполнение работы по подготовке технических носителей информации, обеспечивающих автоматический ввод данных в вычислительную машину, по накоплению и систематизации показателей нормативного и справочного фонда, разработке форм исходящих документов, внесению необходимых изменений и своевременному корректированию рабочих программ;
* участие в выполнении различных операций технологического процесса обработки информации (прием и контроль входной информации, подготовка исходных данных, обработка информации, выпуск исходящей документации и передача ее заказчику);
* ведение учета использования машинного времени, объемов выполненных работ;
* выполнение отдельных служебных поручений своего непосредственного руководителя.

# 4 Техно-рабочий проект

## 4.1 Постановка задачи

Существует несколько типов обучающих программ:

* обучающие, содержащие только теоретический материал;
* тестирующие, проводящие проверку знаний по предложенным темам;
* обучающе-тестирующие, которые содержат теоретический материал и тесты по предложенным темам.

Требуется разработать обучающе-тестирующее средство по белорусской литературе, предназначенное для учащихся младших курсов ССУЗов и позволяющее проверить свои знания по предложенным темам.

Программное средство должно выполнять следующие функции:

* предоставление информации пользователю. Данная функция должна включать:
  + вывод теоретических сведений;
  + вывод тестовых заданий с вариантами ответов;
  + вывод результатов тестирования;
  + вывод статистики;
  + вывод справки.
* ввод информации о пользователе;
* тестирование;
* обработка результатов тестирования.

Разрабатываемое программное средство должно сохранять данные, вводимые пользователем при выборе варианта ответа на задание и должно содержать простой и понятный интерфейс.

## 4.2 Выбор языка программирования

Delphi – язык и среда программирования, относящаяся к классу RAD (Rapid Application Development – средство быстрой разработки приложений) – средств CASE-технологии. Delphi сделала разработку мощных приложений Windows быстрым процессом, доставляющим удовольствие. Приложения Windows, для создания которых требовалось большое количество человеческих усилий, например в C++, теперь могут быть написаны одним человеком, использующим Delphi.

Интерфейс Windows обеспечивает полное перенесение CASE-технологий в интегрированную систему поддержки работ по созданию прикладной системы на всех фазах жизненного цикла работы и проектирования системы.

Delphi обладает широким набором возможностей, начиная от проектировщика форм и кончая поддержкой всех форматов популярных баз данных. Среда устраняет необходимость программировать такие компоненты Windows общего назначения, как метки, пиктограммы и даже диалоговые панели. Работая в Windows, можно видеть одинаковые “объекты” во многих разнообразных приложениях. Диалоговые панели (например, Open File и Save File) являются примерами многократно используемых компонентов, встроенных непосредственно в Delphi, который позволяет приспособить эти компоненты к имеющийся задаче, чтобы они работали именно так, как требуется создаваемому приложению. Также здесь имеются предварительно определенные визуальные и не визуальные объекты, включая кнопки, объекты с данными, меню и уже построенные диалоговые панели. С помощью этих объектов можно, например, обеспечить ввод данных просто несколькими нажатиями кнопок мыши, не прибегая к программированию. Это наглядная реализация применений CASE-технологий в современном программировании приложений. Та часть, которая непосредственно связана с программированием интерфейса пользователя системой, получила название визуальное программирование

Выгоды от проектирования в среде Windows с помощью Delphi:

* устраняется необходимость в повторном вводе данных;
* обеспечивается согласованность проекта и его реализации;
* увеличивается производительность разработки и переносимость программ.

Визуальное программирование как бы добавляет новое измерение при создании приложений, давая возможность изображать эти объекты на экране монитора до выполнения самой программы. Без визуального программирования процесс отображения требует написания фрагмента кода, создающего и настраивающего объект “по месту”. Увидеть закодированные объекты было возможно только в ходе исполнения программы. При таком подходе достижение того, чтобы объекты выглядели и вели себя заданным образом, становится утомительным процессом, который требует неоднократных исправлений программного кода с последующей прогонкой программы и наблюдения за тем, что в итоге получилось.

Благодаря средствам визуальной разработки можно работать с объектами, держа их перед глазами и получая результаты практически сразу. Способность видеть объекты такими, какими они появляются в ходе исполнения программы, снимает необходимость проведения множества операций вручную, что характерно для работы в среде, не обладающей визуальными средствами – вне зависимости от того, является она объектно-ориентированной или нет. После того, как объект помещен в форму среды визуального программирования, все его атрибуты сразу отображаются в виде кода, который соответствует объекту как единице, исполняемой в ходе работы программы.

Размещение объектов в Delphi связано с более тесными отношениями между объектами и реальным программным кодом. Объекты помещаются в вашу форму, при этом код, отвечающий объектам, автоматически записывается в исходный файл. Этот код компилируется, обеспечивая, существенно более высокую производительность, чем визуальная среда, которая интерпретирует информацию лишь в ходе исполнения программы.

Три основные части разработки интерфейса следующие: проектирование панели, проектирование диалога и представление окон. Для общего пользовательского доступа также должны учитываться условия применения архитектуры прикладных систем.

Сегодня появилась реальная возможность с помощью моделирования на современных многофункциональных средствах обработки и отображения информации, таких как Delphi конкретизировать тип и характеристики используемых информационных моделей, выявить основные особенности будущей деятельности операторов, сформулировать требования к параметрам аппаратно-программных средств интерфейса взаимодействия и т.д.

Использование типовых решений, модульного принципа проектирования систем отображения и обработки информации приобретает всё более широкие масштабы, что, впрочем, вполне естественно.

Особый упор при внедрении данных задач следует, конечно, придавать современным CASE-средствам разработки программ, так как они наиболее оптимально позволяют проектировать решения, в основе которых лежат, в первую очередь, требования к согласованному пользовательскому интерфейсу, каковым и является интерфейс Windows. Никакие продукты других фирм, доступные сегодня, не обеспечивают одновременную простоту использования, производительность и гибкость в такой степени, как Delphi. Этот язык заполнил брешь между языками 3-го и 4-го поколений, соединив их сильные стороны и создав мощную и производительную среду разработки.

## 4.3 Проектирование

Одной из главных задач при проектировании пользовательского интерфейса является создание не только функциональной, но и удобной для работы пользователя среды. Интерфейс должен отвечать следующим требованиям:

-понятность для пользователей различной классификации;

-отсутствие необходимости дополнительного обучения.

При запуске программного средства пользователю должна представляться форма с расположенными на ней необходимыми компонентами для работы с программой. На главной форме будет расположено пользовательское меню, с которого и будет начинаться работа пользователя. Меню должно быть сделано таким образом, чтобы пользователь мог одним двумя кликами попасть в любую часть программы и приступить к работе. В связи, с чем предлагается разбить меню на следующие пункты:

* вкладка «Файл» должна будет включать в себя:
  + «Закрыць» – будет использоваться для выхода из программы;
* вкладка «Дзеянне» должна будет включать в себя:
  + «Прайсцi тэст» – для перехода к тестированию;
  + «Паглядзець статыстыку» – для просмотра архива результатов тестирования;
  + «Пачытаць тэорыю» – будет открывать файл с теоретическими сведениями об авторах;
* вкладка «Аб праграме» должна будет включать в себя:
* «Даведка» – для просмотра справочной информации о работе программы;
* «Аб праграме» – будет использоваться для просмотра информации о разработчиках программы.

Большую сложность при проектировании обучающе-тестирующего средства представляет определение различных параметров.

Программное средство будет содержать 14 тестов, каждый из которых соответствует одному из авторов, творчество которых изучается учащимися младших курсов ССУЗов. Тесты будут иметь различное количество вопросов (от 8 до 59) и различное максимальное количество ответов (от 5 до 23). Кроме того, количество символов, необходимое для определения вопросов и ответов тоже разнится, и от этого зависит количество строк, занимаемых вопросами и ответами соответственно.

Для корректного распределения всего необходимого текста по экрану монитора необходима точная информация о количестве строк, занимаемых отдельными «кусками» текста. Необходимо написать отдельную программу, которая будет определять и записывать всю необходимую информацию о параметрах.

Файл с теорией будет создан с использованием языка гипертекстовой разметки. Биография каждого автора будет размещаться в отдельном html-файле. После написания всех необходимых html-файлов, они будут соединены в отдельный файл справки при помощи программы htm2chm.exe. Результатом этой работы будет файл справки theory.chm. Этот файл будет открываться из программного средства каждый раз, когда необходимо будет почитать теорию.

## 4.4 Алгоритм решения задачи

В качестве алгоритма решения задачи будет представлен алгоритм программы, используемой для получения информации о параметрах размещения текста вопросов и ответов тестов на экране монитора. Этот алгоритм иллюстрирован на рисунке 2.

1

5

3

Конец цикла

7

4

6

Конец цикла

8

9

2

10

11

12

13

14

15

19

20

21

25

26

Конец цикла

22

23

24

Конец цикла

27

16

17

18

да

нет

да

да

да

да

да

нет

нет

нет

нет

нет

нет

да

Рисунок 2 – Алгоритм программы для записи параметров размещения текста на экране

Для программы, записывающей параметры размещения текста на экране использованы следующие переменные и структуры данных:

* f – файловая переменная, которая используется для обращения к файлу с тестом;
* i, ii, j, k – переменные счетчики, используемые в циклах;
* maxo – максимальное количество ответов в тесте;
* mo – количество вариантов ответов, предложенных на данный вопрос;
* n – переменная-ограничитель в циклах (количество строк в файле);
* nom – переменная, определяющая номер вопроса;
* ooo – двумерный массив, использующийся для записи количества строк, отводимых каждому варианту ответа всех тестов (строки соответствуют номерам ответов, столбцы – номерам вопросов);
* r – массив строк из файла;
* s – переменная строкового типа, используемая для удобного обращения к отдельной строке;
* vopr – запись, характеризующая отдельный вопрос, состоящая из следующих полей:
  + param – количество строк, отводимых для вопроса;
  + stroka – номер строки в файле, с которой начинается вопрос;
  + kolot – количество вариантов ответа, предлагаемых на данный вопрос.

Описание алгоритма описано в таблице 1.

Т а б л и ц а 1 – Описание алгоритма

|  |  |
| --- | --- |
| Номер блока | Описание блока |
| 1 | Определение и открытие файла для чтения с текстом теста, а также обнуление переменных maxo – максимальное количество ответов, mo, i, n |
| 2 | Цикл пока не конец файла |
| 3 | Чтение и занесение в массив r строк из файла |
| 4 | Закрытие файла, обнуление переменной nom – номер вопроса |
|  | Продолжение таблицы 1 |
| 5 | Цикл пока i<=n |
| 6 | Инкрементация текущего номера i в массиве r, присвоение строковой переменной s элемента массива текущего номера |
| 7 | Cтрока s не пустая и первый символ строки равен «#»? |
| 8 | s не пустая строка? |
| 9 | Инкрементация mo |
| 10 | Инкрементация nom |
| 11 | nom>1? |
| 12 | Присваивание vopr[nom-1].kolot значения mo |
| 13 | maxo<mo? |
| 14 | Присваивание maxo значения mo |
| 15 | Обнуление mo, присваивание vopr[nom].stroka значения i и vopr[nom].param значения parametr(0,s) |
| 16 | maxo меньше mo? |
| 17 | Присваивание maxo значения mo |
| 18 | Присваивание v[nom].kolot значения mo, обнуление ii и k |
| 19 | Цикл для i от 1 до n |
| 20 | Присваивание s значения r[i] |
| 21 | s пустая строка? |
| 22 | Первый символ строки s равен «#»? |
| 23 | Обнуление ii, инкрементация k |
| 24 | Инкрементация ii, присваивание ооo[ii,k] значения parametr(1,s) |
| 25 | Цикл для j от k+1 до maxo |
| 26 | Обнуление oоо[ii,j] |
| 27 | Вывод всех полученных параметров на экран |

## 4.5 Руководство программиста

Работа с программой начинается с запуска приложения проекта Test.exe. На рисунке 3 представлено основное окно программного средства.

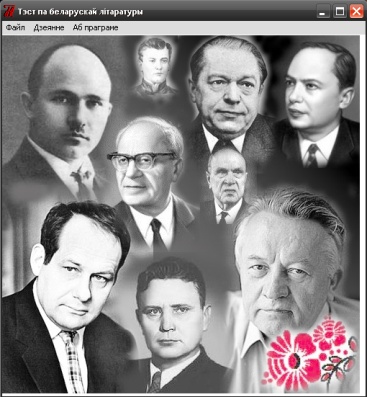


Рисунок 3 – Окно приложения, которое открывается после запуска программного средства

На основной форме программного средства расположено пользовательское меню:

а) вкладка «Файл» включает в себя:

1) «Закрыць» - используется для выхода из программы;

б) вкладка «Дзеянне» включает в себя:

1) «Прайсцi тэст» - используется для перехода к тестирующему блоку;

2) «Паглядзець статыстыку» - используется для просмотра архива результатов тестирования;

в) вкладка «Аб праграме» включает в себя:

1) «Даведка» используется для просмотра справки;

2) вкладка «Аб праграме» используется для просмотра информации о разработчиках программного средства.

Выбрав пункт «Прайсцi тэст» вкладки «Дзеянне» пользователь перейдет к авторизации. Пользователь должен будет ввести данные о себе: имя и номер учебной группы. Авторизация пользователя приведена на рисунке 4.



Рисунок 4 – Авторизация пользователя

После авторизации пользователь перейдет к выбору теста. Всего будет представлено 2 теста. Выбор теста приведен на рисунке 5.



Рисунок 5 – Выбор теста

После выбора одного из тестов пользователь перейдет к выполнению тестирования. Чтобы подтвердить начало тестирования, пользователь должен нажать кнопку «Пачать тесцiраванне». Если пользователь передумал проходить тест, он должен нажать кнопку «Спынiць тэсцiраванне». Подтверждение о начале тестирования представлено на рисунке 6.



Рисунок 6 – Подтверждение начала тестирования

В каждом варианте будут предложены различные задания. В каждом задании по несколько вариантов ответов. Правильными могут быть как один, так и несколько. Выбор варианта ответа будет производиться щелчком левой кнопки мыши по ответу. После выбора варианта ответа пользователь должен будет нажать кнопку «Наступнае пытанне», для того, чтобы перейти к следующему вопросу. Тестирование представлено на рисунке 7.

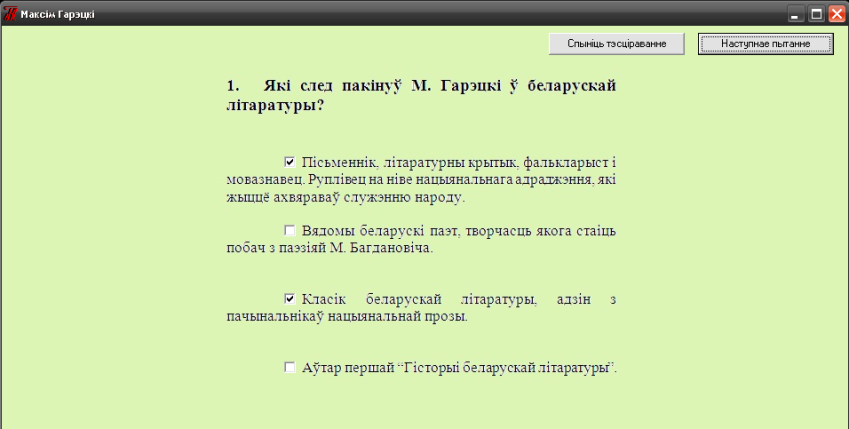


Рисунок 7 – Тестирование

В тестах могут встретиться вопросы, в которых нужно соотнести вопрос и вариант ответа. Напротив вопроса расположен выпадающий список номеров ответов. Пользователю необходимо выбрать правильное, на его взгляд, соотношение. Варианты заданий с соотношением вопросов и ответов представлены на рисунке 8.

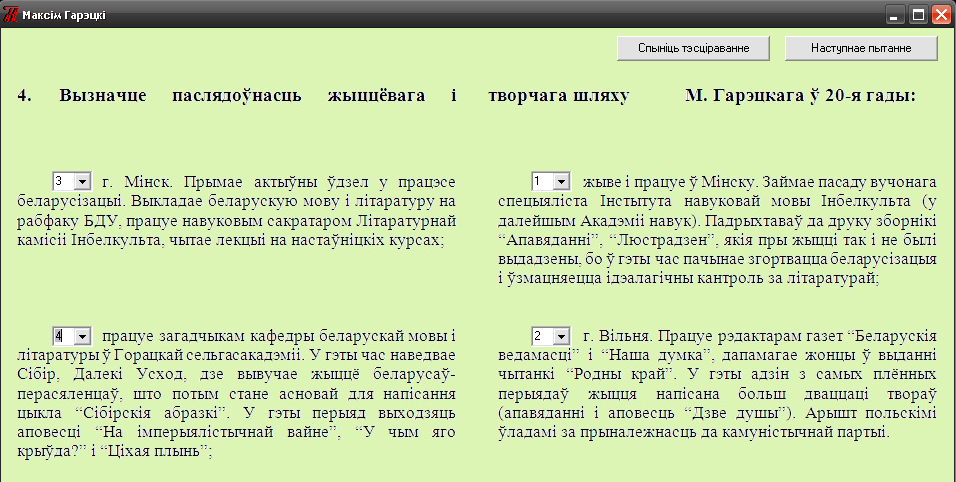


Рисунок 8 – Задание с соотношением вариантов ответов

После завершения тестирования на экран будет выведено сообщение о полученной оценке. Результат тестирования представлен на рисунке 9.

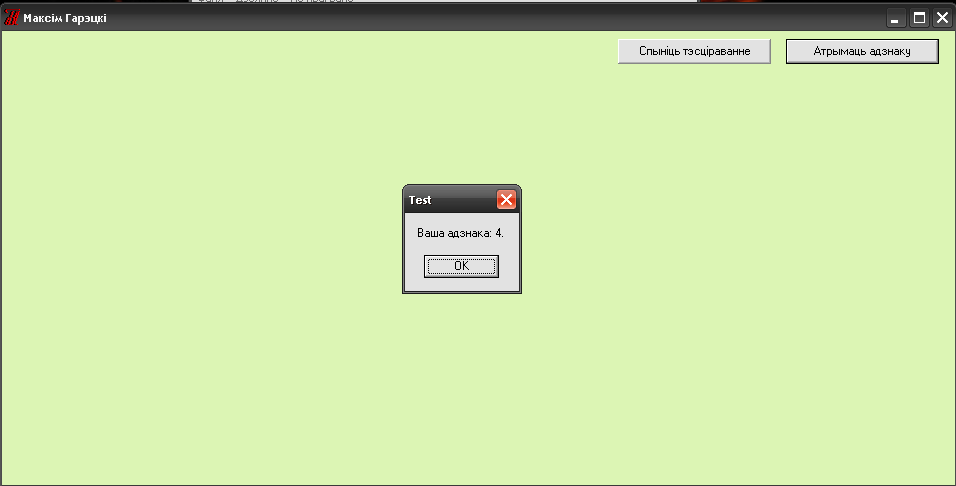


Рисунок 9 – Результат тестирования

Также пользователь сможет просмотреть архив результатов тестирования. В архиве будет представлена следующая информация: имя пользователя, номер учебной группы, название теста и полученная оценка. Статистика представлена на рисунке 10.



Рисунок 10 – Статистика

При выборе пункта меню «Даведка» пользователю будет представлена справка о программе. Справка представлена на рисунке 11.

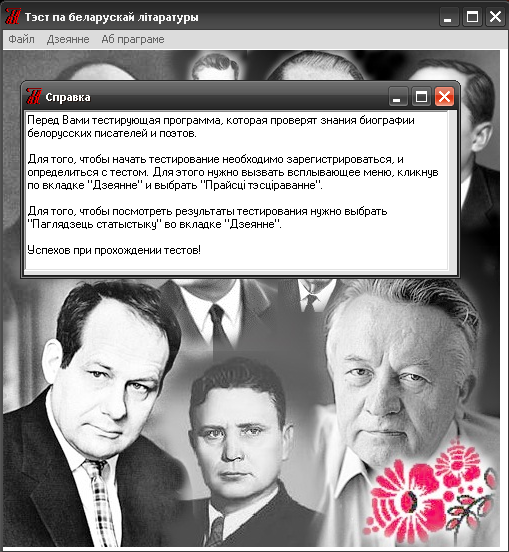


Рисунок 11 – Справка

При выборе пункта меню «Аб праграме» пользователю будет представлена информация о разработчиках данного программного продукта. Результат выбора пункта меню «Аб праграме» представлен на рисунке 12.

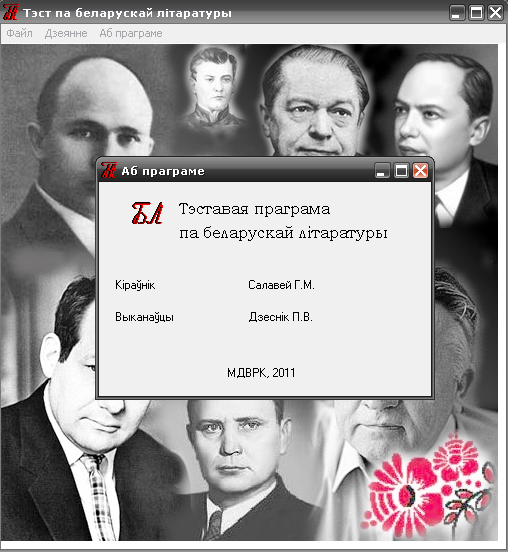


Рисунок 12- Результат выбора пункта меню «О программе»

# Заключение

Подводя итоги работы над разработкой программного средства, можно сделать следующие выводы:

1) поставленная задача выполнена:

а) программное средство позволяет проводить самопроверку по выбранной теме;

б) программное средство предусматривает возможность просмотра архива результатов тестирования;

2) программное средство отлажено и протестировано.

Однако, при тестировании программного средства были выявлены некоторые недостатки, одним из которых является система выставления оценки. Оценка ставится на основе процентного соотношения правильных ответов. Изначально, при запуске, программа формирует две строки, первая из которых состоит из символов соответствующих правильным ответам, причем все части строки, соответствующие своим определенным вопросам, между собой имеют равное количество символов, а вопросы имеют разное количество вариантов ответа. Вторая строка – это строка ответов пользователя. Если в обеих строках на одних и тех же позициях стоят одинаковые символы, то пользователю засчитывается балл. За сто процентов принимается количество символов в строке. Все неотмеченные варианты записываются в строку ответов пользователя символом «0», также записаны неверные варианты в строке правильных ответов. Но из-за того, что фактически вопрос может иметь меньше вариантов ответа, чем это записано, программа засчитывает большее количество правильных ответов, что не соответствует действительности, и пользователь получает большую оценку, чем заслуживает.

Вторым большим недостатком является отсутствие возможности управления статистикой программно. Следовательно, необходимо создать вход администратора.

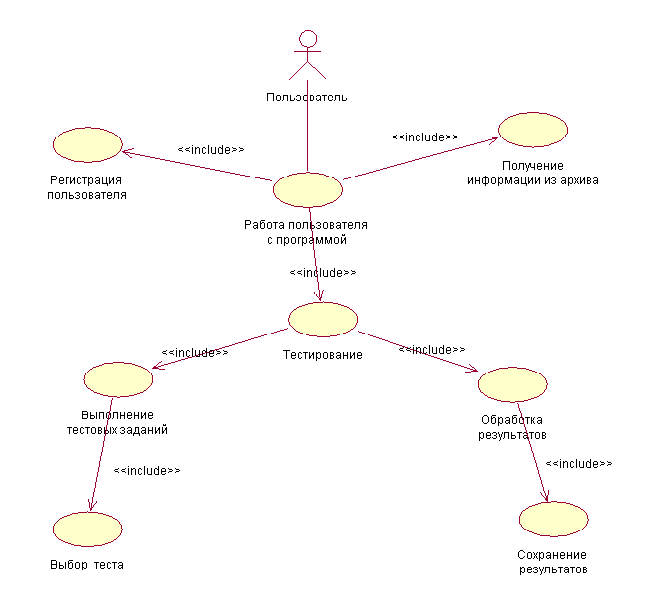
В дальнейшем планируется исправить все недостатки программы, а также разработать большее количество тестов.

# Литература

1. I.Р. Ракей, Н.М. Акулiч Беларуская літаратура ХХ стагоддзя: вучэбна-метадычны дапаможнік. – Мінск: МДВРК, 2010. – 132 с.
2. Бобровский С. И. Delphi 7 Учебный курс. – СПб.: Питер, 2004. – 736 с.: ил.
3. Культин Н.Б. Основы программирования в Delphi7. – СПб.: БХВ – Петербург, 2003. – 608 с.
4. Сухарев М. Delphi. Полное руководство – СПб.: Наука и техника, 2008. – 1040 с.
5. Фаронов В. В. Delphi Программирование на языке высокого уровня: Учебник для ВУЗов – СПб.: Питер, 2008. – 640 с.: ил.
6. Фаронов В. В. Система программирования Delphi. – СПб.: БХВ – Петербург, 2003. – 912 с.
7. Фленов М. Библия Delphi – СПб.: БВХ – Петербург, 2011. – 688 с.
8. Фленов М. Е. Библия Delphi. – СПб.: БХВ-Петербург, 2004. – 880 с.: ил.
9. Хомоненко А. Д. и др. Delphi 7 / Под общ. ред. А.Д. Хомоненко. - СПб.: БХВ-Петербург 2003. - 1216 с.

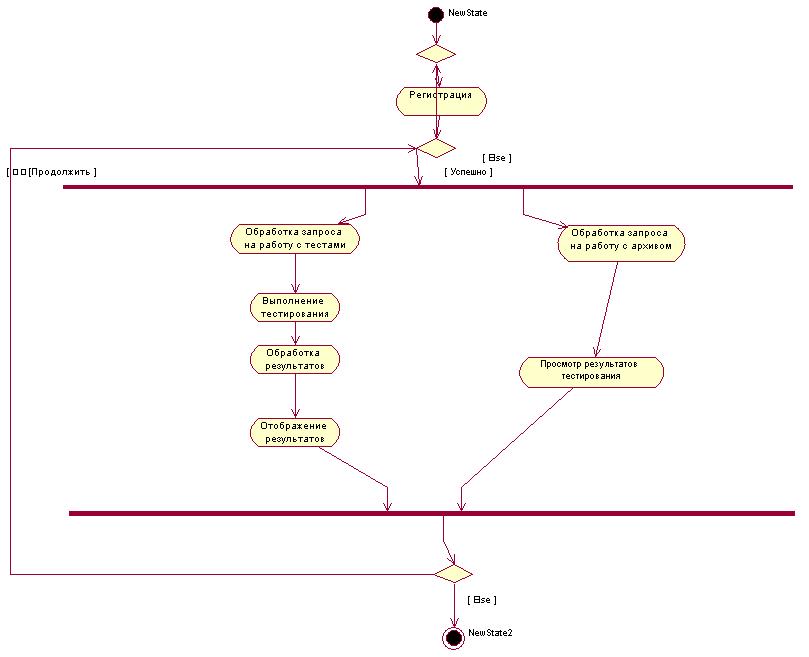
# Приложение А

Диаграмма вариантов использования



# Приложение Б

Диаграмма деятельности



# Приложение В

Листинг

program l1;

type

vopr=record

param,stroka,kolot:integer;

end;

ooo=array[1..30,1..60] of integer;

var

f:text;

s:string;

r:array[1..600] of string;

i,ii,j,n,nom,k:integer;

v:array[1..60] of vopr;

o:ooo;

maxo, mo:integer;

begin

assign(f,'f:\Тест по беллиту\Count\3.txt');

reset(f);

maxo:=0;

mo:=0;

i:=0;

n:=0;

while not(eof(f)) do

begin

readln(f,s);

inc(n);

r[n]:=s;

end;

close(f);

nom:=0;

while i<=n do

begin

inc(i);

s:=r[i];

if (s<>'')and(s[1]='#') then

begin

inc(nom);

if nom>1 then

v[nom-1].kolot:=mo;

if maxo<mo then maxo:=mo;

mo:=0;

v[nom].stroka:=i;

v[nom].param:=parametr(0,s);

end

else

if s<>'' then

inc(mo);

end;

if maxo<mo then maxo:=mo;

v[nom].kolot:=mo;

ii:=0;

k:=0;

for i:=1 to n do

begin

s:=r[i];

if s='' then

for j:=k+1 to maxo do

begin

o[ii,j]:=0;

end

else

begin

if s[1]='#' then

begin

ii:=0;

inc(k);

end

else

begin

inc(ii);

o[ii,k]:=parametr(1,s);

end;

end;

end;

//writeln('Параметры вопросов:');

write('par:array[1..n] of integer=(');

for i:=1 to nom-1 do

write(v[i].param,',');

write(v[nom].param,');');

writeln;

//writeln('Строка:');

write('noms:array[1..n] of integer=(');

for i:=1 to nom-1 do

write(v[i].stroka,',');

write(v[nom].stroka,');');

writeln;

//writeln('Количество ответов:');

write('kolot:array[1..n] of integer=(');

for i:=1 to nom-1 do

write(v[i].kolot,',');

write(v[nom].kolot,');');

writeln;

//writeln('Параметры ответов:');

i:=0;

while i<maxo do

begin

inc(i);

write('oo',i,':array[1..n] of integer=(');

for j:=1 to nom-1 do

write(o[i,j],',');

write(o[i,nom],');');

writeln;

end;

end.