СОДЕРЖАНИЕ

Для третьей лаб работы делаю словно какие-то изменения

[ОБЩИЕ СВЕДЕНИЯ 2](#_bookmark0)

[ОПИСАНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ 2](#_bookmark1)

Требования к структуре и функционированию системы 3

[Другие требования](#_bookmark2) 3

ФУНКЦИОНАЛЬНАЯ СТРУКТУРА СИСТЕМЫ 4

Описание процесса выполнения функции «Добавление ученика» 4

Описание процесса выполнения функции «Добавление учителей по дисциплинам» 5

[Функционально-структурная схема подсистемы](#_bookmark3) 6

Требования к техническому обеспечению 8

[ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ СИСТЕМЫ](#_bookmark4) 9

[ИСТОЧНИКИ РАЗРАБОТКИ 9](#_bookmark5)

[КОНТРОЛЬНЫЕ ВОПРОСЫ 9](#_bookmark5)

ОБЩИЕ СВЕДЕНИЯ

Информационная система учета успеваемости учеников школы. Работа создаётся разработчиком поэтапно в соответствии с календарным планом Проекта. По окончании каждого из этапов работ Разработчик сдает Заказчику соответствующие отчетные документы этапа.

ОПИСАНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

Развитие информационных технологий в такой отрасли непроизводственной сферы, как образование, немаловажно для общества. Это одно из приоритетных направлений государственной политики для достижения социального, образовательного и политического благополучия. Создание информационных управляющих систем на государственных учреждениях способствует улучшению их работы, повышению качества услуг и приводит к сбережению государственных средств, выделяемых для поддержки устаревших информационных систем. Направлена на улучшение функционирования школы.

В отрасли существуют автоматизированные системы работы в школе, отрасль хорошо оснащена компьютерной техникой, однако не столь хорошо автоматизирован учет учеников и всех сопутствующих тому объектов. До сих пор сотрудники подобных подразделений ведут учет вручную в десятках журналах, в которых одна и та же информация повторяется несколько раз, это приводит к увеличению трудоемкости учетных операций за счет выполнения сотрудниками прямых обязанностей.

Целью информационной системы учета является создание не сложной автоматизированной системы отвечающей всем требованиям учета в школе.

Назначением данной подсистемы является учет успеваемости учеников. Учет успеваемости обеспечивает уменьшение трудоемкости работ по учету, получение оперативной информации по различным ученикам и их оценкам.

Требования к структуре и функционированию подсистемы

Учитывая специфику выбранного задания, перечень функций подсистемы автоматизируемых, должен быть сроднившийся функциям, которые автоматизируются в процессе разработки подсистем.

Таким образом, подсистема, разрабатывается, должна выполнять следующие функции:

* ведение сведений о учениках;
* потребность преподавателей по дисциплинам;
* формирование текущего отчета о учениках;
* формирование текущего отчета об успеваемости учеников.

Другие требования

ПО должно быть написано удобно для пользователя любой квалификации. После введения и контроля входящих документов должны формироваться выходные данные, оформленные в соответствии со стандартами. Оболочка базы должна иметь удобный интерфейс доступен обычному человеку не знакомому с данным программным обеспечением. Вносить данные в БД сможет только секретарь со своего рабочего места.

Однако работники, которые будут пользоваться компьютеризированной подсистемой, должны иметь навыки работы на персональном компьютере и в среде Windows. Также они должны владеть навыками работы с периферийными устройствами (принтерами, сканерами), должны иметь опыт работы с Microsoft Office.

ФУНКЦИОНАЛЬНАЯ СТРУКТУРА СИСТЕМЫ

Описание процесса выполнения функции «Добавление ученика»

Эта функция должна обеспечивать внесение в БД информации о учениках.

Доступ к внесению данных должен предоставляться только специалисту БД.

На рисунке 1 отражена диаграмма декомпозиции для функции

«Добавление ученика», которая подробнее описывает данную функцию. На рисунке 1 можно увидеть, что специалист контролирует внесение данных в БД.

Назначение и характеристика с помощью программного обеспечения осуществляется внесением и представлением информации в нужном виде на экран.

Используемая информация: данные полученные от учеников.

Результаты решения: результатами работы являются исходная информация в виде форм базы данных.

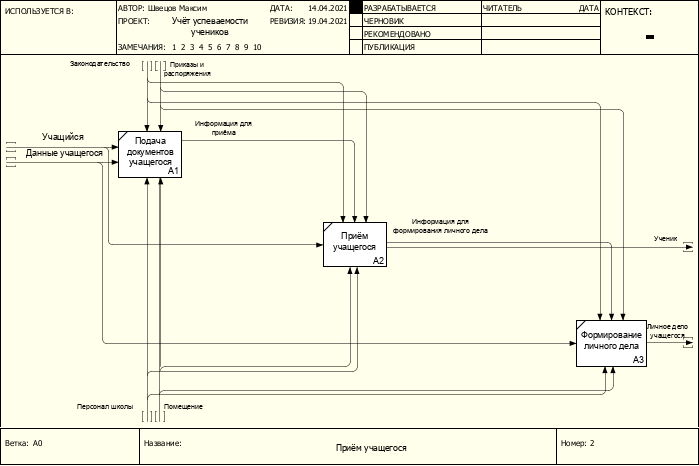


Рисунок 1 – Диаграмма декомпозиции для функции «Добавление ученика»

Описание процесса выполнения функции «Добавление учителей по дисциплинам»

Эта функция должна обеспечивать внесение в БД информации о учителях и дисциплинах, которые они ведут.

Доступ к внесению данных должен предоставляться только специалисту БД.

На рисунке 2 отражена диаграмма декомпозиция для функции

«Добавление учителей по дисциплинам», которая подробнее описывает данную функцию. На рисунке 2 можно увидеть, что специалист контролирует внесение данных в БД.

Назначение и характеристика с помощью программного обеспечения осуществляется внесением и представлением информации в нужном виде на экран.

Используемая информация: данные полученные от учителей.

Результаты решения: результатами работы являются исходная информация в виде форм базы данных.

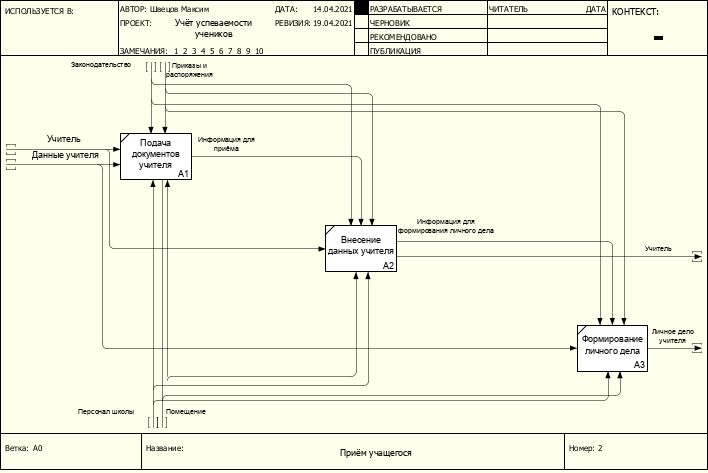


Рисунок 2 – Диаграмма декомпозиции для функции «Добавление учителей по дисциплинам»

Функционально-структурная схема подсистемы

Функционально-структурная схема была построена с помощью case- технологий. CASE-технологиями называют реализованные в виде программных продуктов технологические системы, ориентированные на создание сложных программных систем и поддержку их полного жизненного цикла или его основных этапов.

Для построения функционально-структурной схемы было выбрано программное обеспечение Ramus.

Ramus – это кроссплатформенная система моделирования и анализа бизнес процессов. Данная система предназначена для:

* разработка графических моделей бизнес-процессов (поддерживаются нотации IDEF0 и DFD);
* разработка систем классификации и кодирования (с привязкой к моделям процессов);
* формирование отчётности по моделям и системе классификации.

Преимуществами данного программного обеспечения являются гибкий графический интерфейс пользователя и поддержка неограниченного количества атрибутов различных типов.

В данной работе субъектом будет выступать само учебное заведение, а именно процессы, происходящие внутри него; цель моделирования - воспроизвести процессы, происходящие в учебном. Модель показана на рисунке 4.

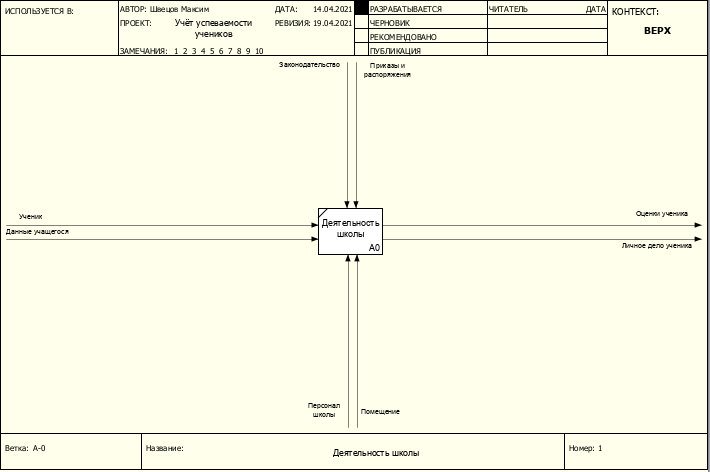


Рисунок 3 - Диаграмма А-0

Для рассматриваемого учебного заведения входными стрелками будут:

* Учащийся;
* Данные учащегося. Выходные стрелки:
  + Личное дело учащегося. Стрелки управления:
  + Законодательство;
  + Приказы и распоряжения. Стрелки механизма:
  + Помещение;
  + Персонал школы.

После построения контекстной диаграммы детализируем её с помощью диаграммы A0. Диаграмма декомпозиции первого уровня показана на рисунке 4.

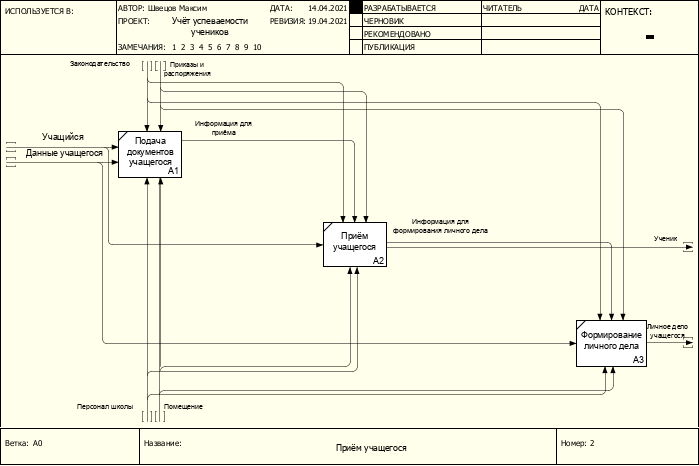


Рисунок 4 - Диаграмма А0

В качестве стрелок механизма для каждой работы будут стрелки

«Персонал школы» и «Помещение».

В качестве стрелок управления для каждой работы будут стрелки

«Законодательство» и «Приказы и распоряжения».

Для обеспечения высокой надежности функционирования системы и ее отдельных компонентов должно обеспечиваться выполнение требований по диагностированию ее состояния. Диагностирование должно осуществляться программными средствами входящими в комплект поставки программного обеспечения.

Требования к техническому обеспечению

Система может быть реализована с использованием ПК работника, который будет осуществлять учёт учеников или же реализована с использованием сервера. ПК рекомендуемо должен иметь следующие показатели: 1. Операционная система Windows (версии 7, 8, 8.1, 10) или Mac OS X Yosemite 10.10 и выше; 2. Оперативная память: 4 Гб и более; 3. Процессор: - Intel Core 2 Duo или более поздней версии с поддержкой SSE2,

- AMD Athlon X2 и выше; - 2 ядра и выше; - 1,8. Если сервер, то он должен быть развернут на IBMPower 595 Server 9119-FHA. Сервер сбора, обработки и загрузки информации должен быть развернут на ARTLINE Business R27 v10.

ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ СИСТЕМЫ

При приеме заказчику передается инструкция по эксплуатации системы и проверяется функционирование автоматизированной информационной системы на реальных данных, подготовленных и введенных заказчиком по имеющейся инструкции, или, в случае неготовности заказчика, на контрольном примере. Готовый программный продукт подвергается многократному тестированию. Работы по сдаче проводятся на технических средствах разработчика. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие. Полученный программный продукт должен соответствовать разработанному ТЗ и ТП.

ИСТОЧНИКИ РАЗРАБОТКИ

Настоящее техническое задание разработано на основе следующих документов и информационных материалов:

1. Официальные рекомендации P 50.1.028 - 2001 ГОСТ России по применению стандартов IDEF для функционального моделирования.
2. Организационная структура предприятий. Коноков Д.Г., Рожков М.А., Смирнов А.О., Яниковская О.Н., издание второе, Москва, 1999. 176 с.

КОНТРОЛЬНЫЕ ВОПРОСЫ

* 1. Процесс разработки состоит из множества подпроцессов, или дисциплин,некоторые из которых перечислены ниже. Процесс — совокупностьвзаимосвязанных или взаимодействующих видов деятельности,преобразующих входы в выходы.
     + Анализ требований → Спецификация программного обеспечения
     + Проектирование программного обеспечения
     + Программирование
     + Тестирование программного обеспечения
     + Системная интеграция (System integration)
     + Внедрение программного обеспечения (или Установка программного
     + обеспечения)
     + Сопровождение программного обеспечения
  2. Артефакт - это один из многих видов материальных побочных продуктов,возникающих в процессе разработки программного обеспечения. Некоторыепомогают описать функцию, архитектуру , и дизайн программногообеспечения. Другие артефакты связаны с самим процессом разработки,например планы проектов, бизнес-модели и оценки рисков.
  3. Стратегии разработки ПО можно подразделить на три группы:

1. Линейная последовательность этапов разработки – однократный проход(водопадная стратегия);
2. Инкрементная стратегия, когда сначала определяются все требования(пользовательские и системные), а затем оставшаяся часть разработкивыполняется в виде последовательности версий, первая из которых реализуетчасть запланированных возможностей, а все последующие версии реализуютдополнительные возможности до тех пор, пока не будет получена полнаясистема;
3. Эволюционная стратегия. При этой стратегии начальный этап не содержитполного объема требования, они уточняются в ходе разработки новыхпоследовательных версий.
   1. Общие принципы разработки программного обеспечения
      * Частотный принцип
      * Принцип модульности
      * Принцип функциональной избирательности
      * Принцип генерируемости
      * Принцип функциональной избыточности
      * Принцип «по умолчанию»
   2. Сосав персонала разработки ПО:
      * Архитектор;
      * Аналитик;
      * Администратор;
      * Разработчики.
   3. Основные характеристики качества программного обеспечения согласно

стандарту ISO/IEC 25010:2011:

1. Функциональность. ПО признается функциональным, если выполняет возложенные на него задачи, отвечает заданным потребностям пользователей. Данный аспект предполагает правильную и точную работу, совместимость всех входящих в состав компонентов.
2. Надежность. Под надежностью ПО понимают бесперебойное выполнение

возлагаемых на него задач на заданных условиях в течение установленного времени.

1. Юзабилити (удобство использования). Этот параметр характеризует

степень удобства ПО для пользователей, его наглядность, легкость эксплуатации и изучения.

1. Эффективность. Параметру соответствует степень обеспечения продуктом необходимой производительности при заданных условиях.
2. Удобство сопровождения. Этот показатель характеризует простоту анализа, тестирования, коррекции компонентов ПО, его обслуживания, а также степень адаптации к новым условиям.
3. Портативность. Степень легкости его переноса на другую

платформу.

Обеспечение качества ПО предполагает его проверку по каждому из перечисленных параметров, выявление слабых сторон и устранение

неисправностей.

1. Совместимость. Способность программных компонентов взаимодействовать друг с другом.
2. Защищенность, т.е. минимизация угроз, связанных с несанкционированным чтением, изменением информации и т. д. Угрозы могут быть также связаны с некорректным использованием ПО, внешним воздействием со стороны посторонних лиц, выходом из строя технических средств.
3. Инспектирование программ – это просмотр и проверка программ с целью обнаружения в них ошибок.

Для начала процесса инспектирования программы необходимы следующие условия: наличие точной спецификации кода (без полной спецификации невозможно обнаружить дефекты в проверяемом программном компоненте);

члены инспекционной группы должны хорошо знать стандарты разработки; в распоряжении группы должна быть синтаксически корректная последняя версия программы (нет смысла рассматривать код, который «почти pавершен»).

Сам процесс инспектирования должен быть относительно коротким (не более двух часов) и сосредоточенным только на выявлении дефектов, аномалий и несоответствий стандартам. Инспекционная группа не должна предлагать способы исправления дефектов или рекомендовать какие-либо изменения в других программных компонентах.

После инспектирования автор изменяет программу, исправляя обнаруженные ошибки. На этапе доработки координатор принимает решение о том, необходимо ли повторное инспектирование. Если повторное инспектирование не требуется, все обнаруженные дефекты фиксируются документально.

1. Варианты использования это - описание последовательности действий, которые может осуществлять система в ответ на внешние воздействия пользователей или других программных систем. Варианты использования отражают функциональность системы с точки зрения получения значимого результата для пользователя, поэтому они точнее позволяют ранжировать функции по значимости получаемого результата.
2. С-Требования

Требования заказчика к системе фиксируются разработчиками посредством проведения специально организованного опроса-интервью. Но перед этим необходимо идентифицировать пользователей системы, т.е. указать категории лиц, которые должны или гипотетически могут в настоящий момент или в будущем воспользоваться разрабатываемой системой. В зависимости от масштабов системы эта задача может оказаться не тривиальной, поэтому требует особого внимания, поскольку при создании системы необходимо учесть требования всех заинтересованных лиц.

1. D-Требования

Разработчикам программного обеспечения нужна база для проектирования и разработки. Эта база представляет собой детальные требования (D- требования). D-требования состоят из полного списка конкретных свойств и функциональности, которую должна иметь программа. Каждое из этих требований пронумеровано, помечено и отслеживается по ходу разработки. D-требования должны быть согласованы с С-требованиями.