

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра Математического и компьютерного моделирования

Информационная система

«Распределение учебной нагрузки на кафедре»

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса 441 группы

направление 09.03.03 – Прикладная информатика

механико-математического факультета

Жиц Марии Григорьевны

Научный руководитель
доцент, к. ф. – м. н.

А. А. Орёл

Зав. кафедрой
д.ф. – м. н.

Ю.А. Блинков

Саратов 2017

ВВЕДЕНИЕ

Проектирование информационных систем является достаточно сложным процессом. Для разработки информационных систем в настоящее время применяется множество языков программирования, одним из которых является язык программирования Python, который удобен в использовании и предоставляет широкие возможности для решения различных задач в ходе разработки той или иной информационной системы.

Актуальность данной темы обусловлена тем, что в современном обществе неотъемлемую часть занимает стремительная информатизация и широкий спектр информационных технологий. Развитие последних привело к тому, что технологии внедряются во все сферы деятельности. Образование - не исключение. Новейшие информационные технологии в обучении позволяют активнее использовать научный и образовательный потенциал ведущих университетов и институтов, привлекать лучших преподавателей к созданию курсов дистанционного обучения, расширять аудиторию обучаемых.

Цель работы заключается в разработке информационной системы «Распределение учебной нагрузки на кафедре» с применением средств функционального программирования.

Основным результатом работы является построенная для заказчика база данных и разработанный на функциональном языке программирования Python код, содержащий графический интерфейс, с помощью которого можно упростить взаимодействие пользователя и системы с целью получения искомого результата.

Выпускная квалификационная работа делится на следующие разделы: введение, 4 раздела, заключение, а также список используемых источников и два приложения. В первом разделе содержится общая теория информационных систем. Второй раздел - также теоретический, содержащий в себе общие понятия из области проектирования информационных систем, а также про предметную область «Распределение учебной нагрузки на кафедре». В третьем разделе находятся базовые UML- и ER-диаграммы, построенные для разработки информационной системы, а также код создания базы данных. Четвертый раздел рассказывает об основах программирования на языке Python и содержит программную реализацию информационной системы на

языке программирования Python. В заключении подводятся итоги проведенной работы.

Основное содержание работы

Во введении описываются цель работы, а также средства, необходимые для грамотного проектирования и дальнейшей реализации информационной системы. В данном случае это:

- язык моделирования UML;
- язык баз данных SQL и СУБД SQLite;
- платформа для разработки приложений Qt;
- язык программирования Python.

В первом разделе содержится общая теория информационных систем.

В частности, основные решаемые задачи:

- Анализ потоков разнообразной информации, перемещающихся в обществе, и их прогнозирование.

Здесь анализируются потоки документов для выявления возможностей их минимизации, стандартизации и приспособления для эффективной обработки на вычислительных машинах; изучаются также особенности потоков информации, проходящей через журналы, газеты, радио- и телевизионные каналы и иные каналы распространения информации; оценивается влияние распространяемой информации на научно-технический прогресс и состояние общества в его различных аспектах.

- Исследование способов представления и хранения информации, создание специальных языков для формального описания информации различной природы, разработка специальных приемов сжатия и кодирования информации, аннотирования объемных документов и реферирования их.

В рамках этой задачи осуществляются работы по созданию банков данных большого объема, хранящих информацию из различных областей знаний в форме, доступной для вычислительных машин.

- Построение различных процедур и технических средств для их реализации, с помощью которых можно автоматизировать процесс извлечения

информации из документов, не предназначенных для вычислительных машин, а ориентированных на восприятие их человеком.

Такие исследования связаны, главным образом, с проблемой определения и извлечения основного смысла (содержания) тех или иных документов при вводе их в банки данных и другие информационные хранилища, ориентированные на компьютеры.

- Создание информационно-поисковых систем, способных воспринимать запросы к информационным хранилищам, сформулированные на естественном языке, а также на специальных языках запросов для систем подобного типа.
- Создание сетей хранения, обработки и передачи информации, в состав которых входят информационные банки данных, терминалы, обрабатывающие центры и средства связи.

Все вышеперечисленные работы опираются, с одной стороны, на исследования в прикладной лингвистике, создающей языки для записи информации и поиска ответов в информационных массивах по поступающим запросам, а с другой стороны, на теорию информации, поставляющую модели и методы, используемые при организации циркуляции информации в каналах передачи данных.

Во втором разделе находится теоретическая информация о проектировании информационных системах и описание предметной области. В теории рассматриваются основные подходы к проектированию. Например:

Любой процесс проектирования информационной системы начинается с определения цели проекта. На момент внедрения системы в производство и в течение всего времени ее эксплуатации проект должен обеспечивать:

- требуемую функциональность системы и степень адаптации к изменяющимся условиям ее функционирования;
- безотказную работу системы в требуемом режиме, иными словами - готовность и доступность системы для обработки запросов пользователей;
- простоту эксплуатации и поддержки системы;
- требуемое время реакции системы на запрос;
- требуемую пропускную способность системы;

- необходимую безопасность от утечки данных и несанкционированного использования;
- высокую производительность и, следовательно, эффективность системы.

Затем следует описание предметной области, которая содержит в себе информацию о базовых сущностях системы, а также необходимые процессы для реализации. В распоряжении сотрудника имеются сведения о преподавателях кафедры, которые включают в себя, кроме с анкетных данных, сведения об их ученой степени, занимаемой административной должности и стаже работы. Преподаватели кафедры должны обеспечить проведение занятий по некоторым предметам. По каждому из них существует определенное количество часов. В результате распределения нагрузки получается информация следующего рода: «Преподаватель N проводит занятия по предмету M с группой Z». Нужно также учитывать, что все проводимые занятия делятся на лекционные и практические. По каждому виду занятий устанавливается также свое количество часов.

В третьем разделе находятся непосредственно результаты проектирования информационной системы. Рассматриваются основные UML-диаграммы и ER-диаграмма.

UML - унифицированный язык моделирования, применяемый для объектного моделирования в области разработки программного обеспечения, системного проектирования и отображения организационных структур.

Как правило диаграммы (за исключением самых простых моделей) дают свернутое представление об элементах разрабатываемой системы. Один и тот же элемент может присутствовать либо во всех диаграммах, либо только в нескольких или не присутствовать ни в одной.

Диаграммы могут содержать любые комбинации сущностей, однако на практике применяется сравнительно небольшое количество типовых комбинаций, соответствующих одному из пяти наиболее необходимых видов, которые составляют архитектуру программной системы.

В данной работе представлены следующие диаграммы, которые были построены с помощью сервиса PlantUML Web Server.:

1. Диаграмма вариантов использования;

2. Диаграмма деятельности;
3. Диаграмма классов.

В диаграмме вариантов использования каждый вариант использования соответствует отдельному сервису, который предоставляет моделируемая сущность по запросу актера, то есть определяет способ применения этой сущности. Сервис, который инициализируется по запросу актера, представляет собой законченную неделимую последовательность действий, т.е. по завершении обработки запроса система должна возвратиться в исходное состояние, чтобы быть готовой к выполнению следующих запросов.

Диаграммы деятельности позволяют реализовать в языке UML особенности процедурного и синхронного управления, обусловленного завершением внутренних деятельностей и действий. Основным направлением использования диаграмм деятельности является визуализация особенностей реализации операций классов, когда необходимо представить алгоритмы их выполнения.

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать различные взаимосвязи между отдельными сущностями предметной области, например, такими как объекты и подсистемы, а также описывать их внутреннюю структуру и типы отношений.

В качестве примера в соответствии с рисунком 1 представлена диаграмма вариантов использования.

Диаграмма вариантов использования для ИС
Распределение учебной нагрузки на кафедре

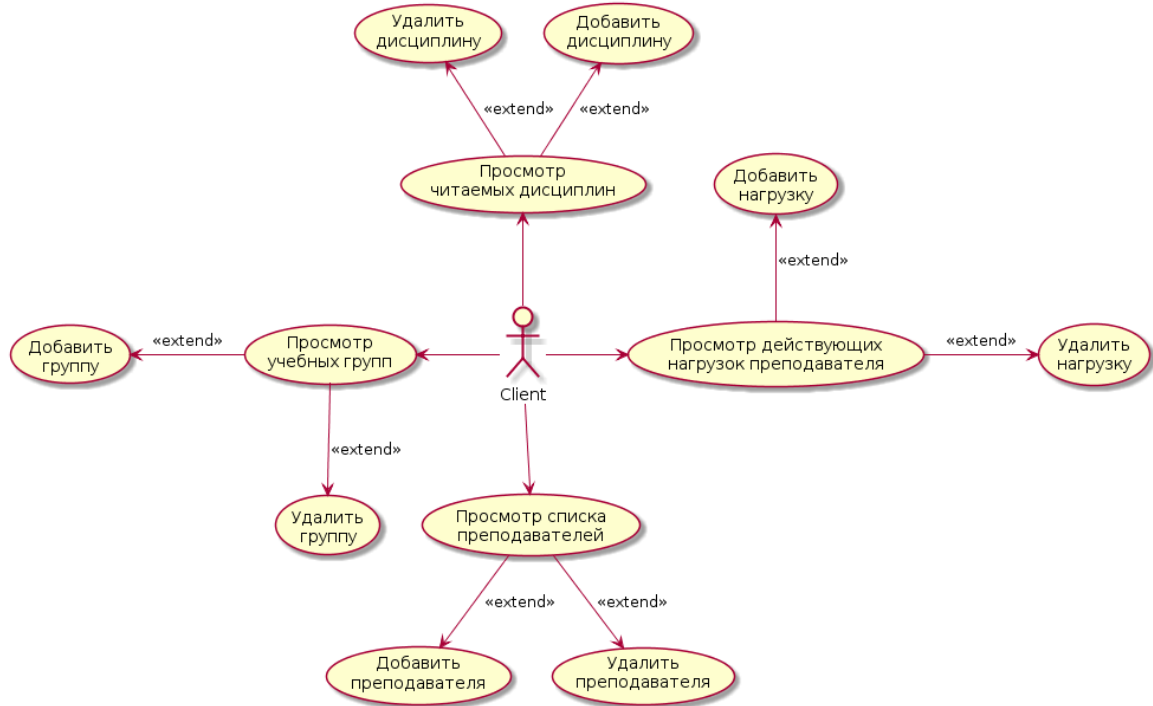


Рисунок 1 — Диаграмма прецедентов

База данных

Также была разработана база данных на языке SQL. SQL — язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных. Язык SQL представляет собой совокупность операторов, инструкций, вычисляемых функций.

В рассматриваемой информационной системе база данных содержит полную информацию о преподавателях, преподаваемых дисциплинах, группах, в которых преподаватели могут вести учебные занятия.

Были выделены основные сущности:

1. «Преподаватели» - фамилия, имя, отчество, пол, должность, ученые степень и звание;
2. «Дисциплины» - наименование дисциплины;
3. «Группы» - номер группы, направление, год приема;

Для написания БД был выбран язык SQL - формальный язык программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей си-

стеймой управления базами данных. Разработанная база данных состоит из 13 таблиц. Среди них можно выделить основные таблицы:

1. «Преподаватель»,
2. «Ученое звание»,
3. «Должность»,
4. «Ученая степень»,
5. «Группа»,
6. «Направление»,
7. «Учебный план»,
8. «Форма аттестации»,
9. «Тип нагрузки»,
10. «Дисциплина».

Данная диаграмма выглядит в соответствии с рисунком 2.

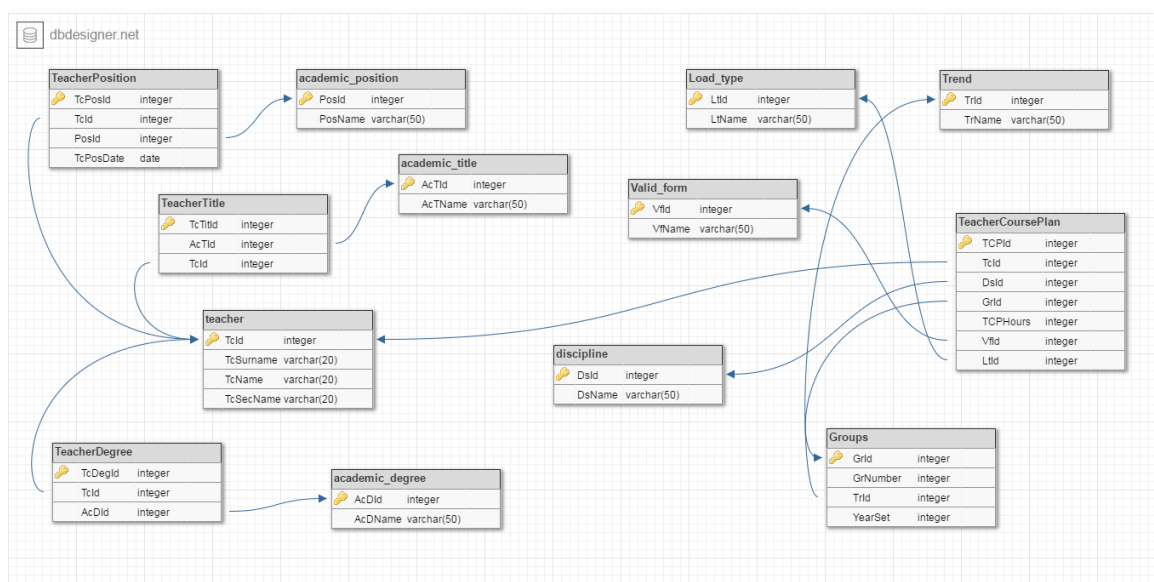


Рисунок 2 — ER-диаграмма

Заполнение таблиц на языке SQL представлено в **приложении А**.

В четвертом разделе находится два подраздела: первый, в котором рассказывается про язык программирования Python, и второй, где описана непосредственно реализация.

Первый подраздел содержит информацию про язык программирования Python и платформу Qt, благодаря которой реализуется пользовательский интерфейс. Например, информация о том, что Qt – одна из ведущих плат-

форм для разработки приложений с графическим пользовательским интерфейсом (GUI) под большинство существующих ныне операционных систем. Также Qt – одноименный набор библиотек, лежащий в основе платформы. А также информация про язык Python и его основные свойства, такие, как интерпретируемость, динамическая типизируемость и легкость в расширении. Работа по реализации ИС осуществляется с помощью подключаемой библиотеки PySide, которая предоставляет расширение языка Python для грамотного использования ООП-идеологии.

Важным качеством как библиотеки Qt, так и интерпретатора Python является их кроссплатформенность, которая позволяет переносить разработанные на PySide приложения из одной операционной системы в другую без каких-либо изменений.

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. Однако стандартная библиотека включает большой объём полезных функций.

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Код в Python организовывается в функции и классы, которые могут объединяться в модули (они в свою очередь могут быть объединены в пакеты).

Использование языка Python позволяет на порядок ускорить процесс разработки приложения для Qt по сравнению с альтернативной разработкой на C++. Достигается это за счет интерпретируемости Python-программы (то есть отсутствует цикл компиляции-сборки), также за счет более высокого уровня языка, что позволяет в несколько раз сокращать объем пишущегося кода.

Реализация ИС, о которой рассказывает второй подраздел, подразумевая, что приложение будет применяться на нескольких ПК, располагающихся в одной локальной сети. Для более удобного функционирования реализован

графический интерфейс для системы. В соответствии с рисунком 3 выглядит список преподаваемых дисциплин на кафедре.

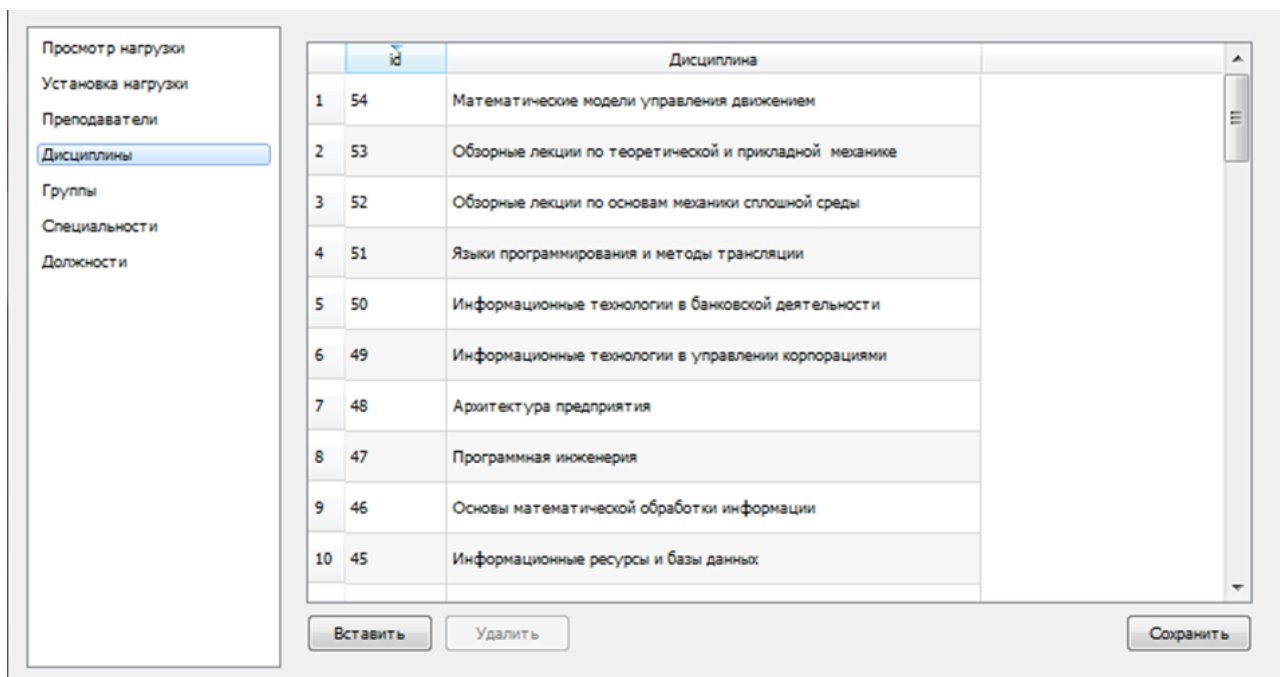


Рисунок 3 — Дисциплины на кафедре

Данный фрагмент реализован следующим фрагментом кода:

```
class PageDiscipline(PageBtn):
    def __init__(self, parent=None):
        model = QSql.QSqlTableModel()
        model.setTable('discipline')
        model.setHeaderData(0, QtCore.Qt.Horizontal, "id", QtCore.Qt.DisplayRole)
        model.setHeaderData(1, QtCore.Qt.Horizontal, "Дисциплина",
            QtCore.Qt.DisplayRole)

        model.setEditStrategy(QSql.QSqlTableModel.OnManualSubmit)

        super(PageDiscipline, self).__init__(model, appname, parent)
```

Полный программный код на языке Python приведен в **приложении Б**.

ЗАКЛЮЧЕНИЕ

Таким образом, в ходе работы были исследованы требования к информационной системе «Распределение учебной нагрузки на кафедре» и спроектирована сама система.

Цели, поставленные во введении, достигнуты и система соответствует требованиям, которые предъявляются к проектированию и разработке информационных систем, а именно:

- требуемая функциональность системы и степень адаптации к изменяющимся условиям ее функционирования;
- безотказная работа системы в требуемом режиме;
- простота эксплуатации;
- высокая производительность.

На практике изученные способы разработки информационной системы были реализованы с помощью программного кода, написанного на высокоуровневом языке программирования Python. Программная реализация также была приложена к данной бакалаврской работе.

Подводя итоги, можно сделать вывод о широких возможностях языка программирования Python для проектирования информационных систем, т.к. он обладает рядом существенных достоинств, например, неизменностью значений переменных.

Все проведенные в работе исследования позволили изучить процесс проектирования информационных систем, функциональное программирование, а также повысить свои навыки решения задач в данных областях как теоретическим, так и практическим способами.