

Передача информации

Любой процесс передачи информации можно представить с помощью схемы: Источник информации → информационный канал → приёмник информации. В передаче информации всегда участвуют две стороны: тот, кто передаёт информацию (источник информации) и тот, кто её получает (приёмник информации).

Человек	Компьютер
Органы чувств человека выполняют роль биологических информационных каналов. Техническими информационными каналами являются телефон, радио, телевидение, компьютер, с помощью которых люди обмениваются информацией.	Наблюдая за природой, используя её идеи, человек создал различные технические устройства для передачи информации: летательный аппарат, навигационные устройства, средства локации, ориентации и связи. Компьютер передаёт информацию с помощью модема, колонок, принтера, плоттера, графопостроителя.

Обработка информации

Компьютер и человек выполняют одни и те же действия с информацией. Но есть существенная разница. Человек это делает сам добровольно или принудительно, а компьютер только с помощью программ, сделанных человеком.

Выводы по практической части

В ходе нашего опроса мы выяснили, что преимущественно наши студенты понимают и разбираются в компьютере. Многие отвечали на вопросы правильно, но и было такое, что ответы были не правильными.

Литература

1. Алексеев, А.Г. Архитектура персонального компьютера [Текст]/А.Г.Алексеев. – М.: АСТ-Пресс, 2018. – 230 с.
2. Бабушкин, А.Н. Что такое персональный компьютер [Текст] /А.Н. Бабушкин. – Воронеж: Велби, 2019. – 458 с.
3. Жигарев, А.Н. Основы компьютерной грамоты [Текст] / А.Н. Жигарев. – СПб.: Машиностроение, 2018. – 520 с.

ПРИМЕНЕНИЕ ENTITY FRAMEWORK К НОРМАЛИЗОВАННОЙ БАЗЕ ДАННЫХ

Никитин Д.А.

*учащийся 4 курса Оршанского колледжа ВГУ имени П.М. Машерова,
г. Витебск, Республика Беларусь*

Научный руководитель – Юржиц С.Л., преподаватель

При проектировании баз данных различного рода приложений рассматривается такой аспект, как приведение базы данных к третьей нормальной форме, и есть проблема в разработке программных продуктов, разрабатываемых на языке программирования C#, с использованием технологии Entity Framework.

Третья нормальная форма – это подход к проектированию схемы базы данных для реляционных баз данных, который использует принципы нормализации для уменьшения дублирования данных, предотвращения аномалий данных, обеспечения ссылочной целостности и упрощения управления данными [2].

Entity Framework – объектно-ориентированная технология доступа к данным, является object-relational mapping (ORM) решением для .NET Framework от Microsoft [3].

Технологию Entity Framework удобно использовать в проектах, где необходимо работать со множеством объектов, т.е. таких проектах как магазины, программы складского учёта и др.

Проблема использования этой технологии появляется при работе с большим количеством сущностей. В ходе работы с базами данных, в частности, в рамках выполнения курсовых проектов, заданий практик приходилось работать с БД с большим количеством сущностей. При приведении базы данных к третьей нормальной форме количество сущностей возрастало, также увеличивалось количество зависимостей между сущностями. Возникла проблема в рациональности приведения отношений базы данных к третьей нормальной форме со стороны разработчика программного средства, а также на каком этапе лучше разрабатывать базу данных – эта особенность относится к технологии Entity Framework, т.к. для её работы необходимо реализовать так называемые модели на языке программирования.

Материал и методы. При рассмотрении проблемы были использованы: теоретические сведения о технологии работы с базами данных Entity Framework; теоретические сведения о приведении отношений базы данных к третьей нормальной форме; практический опыт разработки программного средства с использованием как базы данных в 3НФ, так и технологии Entity Framework.

Результаты и их обсуждение. При разработке программ встал вопрос о том, в какой момент необходимо разрабатывать базу данных. По правилам Entity Framework необходимо разработать модели в виде классов и только после этого создать базу данных, при этом база данных создаётся автоматически на основе описанных классов. Недостатком такого подхода вытекает отсутствие наглядности и визуализации, которую может предоставить СУБД в виде диаграммы связей, а также необходимость установки первичных и внешних ключей вручную. Исходя из вышеописанного разработка базы данных через написание кода не совсем наглядна и может быть сложна с большими объёмами моделей. Также немаловажным фактором является план проектирования, который подразумевает уже на начальных этапах наличие спроектированной базы данных.

На основе недостатков разработки базы данных в среде разработки программного средства был выбран способ создания базы данных в СУБД. Такой способ также имеет недостатки, т.к. все равно необходимо реализовать модели на языке программирования.

Главной проблемой является вопрос о том, в каком из слоёв архитектуры программы выполнить подхват всех зависимостей модели и выбор данных, для полного её формирования. При рассмотрении этого вопроса было выделено два способа: выгрузка данных в необходимый момент в слое, который отвечает за взаимодействие с пользовательским интерфейсом; выгрузка данных при создании модели.

Оба способа подразумевают необходимость расширения базового функционала модели, с соответствующим ограничением этого функционала и свойств модели от базы данных, такое ограничение можно реализовать как при помощи встроенных технологий языка, так и несколькими способами, которые предлагает Entity Framework.

Способ выгрузки данных только в необходимый момент имеет весомое преимущество – нет необходимости перегружать программу данными, которые могут и не понадобиться в процессе использования программы. К недостаткам такого способа можно отнести дублирование кода, этот недостаток перекрывает все преимущества данного способа.

Остаётся способ выгрузки данных при создании модели, его преимуществом является единовременная выгрузка данных из БД. Недостатком этого метода может быть большая нагрузка в момент обращения к базе данных с большим количеством запросов. Также было принято решение разделить выгрузку данных на два этапа: в момент выгрузки всей информации из БД делать выборку только первичных ключей, такой подход на программном уровне позволит избежать передачи большого количества параметров в конструкторе объекта; после выгрузки первичных ключей на их основе подгрузить данные в модели. Такое разделение в

теории должно снизить нагрузку на узел, по которому происходит передача данных. Также для отсутствия зависимостей программ можно использовать потоки для каждой из моделей.

Заключение. Таким образом, в ходе исследования были рассмотрены способы взаимодействия с базой данных, приведённой к третьей нормальной форме с использованием технологии Entity Framework. В ходе разработки выделены два варианта, определяющие момент начала разработки базы данных, рассмотрены преимущества и недостатки обоих методов при разработке. Чаще использовался способ разработки базы данных в СУБД, однако в иных условиях он может оказаться не самым рациональным вариантом. Также были рассмотрены возможные методы выгрузки данных из БД в модели с подхватом имеющихся зависимостей. На основе ряда преимуществ был выбран способ расширения функционала модели, а также выгрузка данных в два этапа.

Литература

1. Руководство по Entity Framework Core [Электронный ресурс] / Сайт о программировании METANIT.COM – Режим доступа: <https://metanit.com/sharp/entityframeworkcore/> – Дата доступа 27.02.2021.
2. Третья нормальная форма [Электронный ресурс] / Свободная энциклопедия Википедия – Режим доступа: https://ru.qaz.wiki/wiki/Third_normal_form – Дата доступа: 10.02.2021.
3. ADO.NET Entity Framework [Электронный ресурс] / Свободная энциклопедия Википедия – Режим доступа: https://ru.wikipedia.org/wiki/ADO.NET_Entity_Framework – Дата доступа: 08.02.2021.

ИСПОЛЬЗОВАНИЕ ПРОГРАММЫ MULTISPEC ДЛЯ АНАЛИЗА И ДЕШИФРИРОВАНИЯ КОСМИЧЕСКИХ СНИМКОВ

Попченко Л.А.

*студент 2 курса ГГУ имени Франциска Скорины,
г. Гомель, Республика Беларусь*

Научный руководитель – Соколов А.С., старший преподаватель

В настоящее время материалы дистанционного зондирования Земли являются одним из основных источников получения информации о текущем состоянии географических объектов, процессов и явлений. Стало доступным бесплатное получение космических снимков со спутников Sentinel-2, Landsat-8 и ряда других. Развитие геоинформационных систем сделало доступным мощный инструментальный анализ и дешифрирования космических снимков, в том числе в бесплатных программах QGIS и MultiSpec. Это сделало возможным широкое применение материалов дистанционного зондирования Земли в преподавании различных геологических и географических дисциплин. Программа доступна для свободного бесплатного скачивания на сайте <https://engineering.purdue.edu/~biehl/MultiSpec/> для Macintosh и Windows.

Целью работы является описание возможностей программы MultiSpec, которые могут быть использованы в образовательном процессе при проведении практических и лабораторных работ.

Результаты и их обсуждение. Космический снимок представляет собой набор изображений территории съёмки, выполненных в разных зонах спектра.

Необходимо загрузить в программу последовательно все изображения с одинаковым пространственным разрешением, затем визуализировать в виде цветного изображения (команда **Processor > Display image**), предварительно задав три изображения, которым будут присвоены синий, зелёный и красный цвета (раздел **Channels** окна **Set Display Specifications for**). Меняя комбинацию каналов съёмки относительно трех каналов экранного изображения (Red, Green, Blue), построить разнообразные цветовые схемы для лучшей визуализации тех или иных классов объектов и явлений [1].

В этом же окне (раздел **Enhancement**) можно реализовать и другие возможности управления визуализацией многозонального изображения: