|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»  *Факультет социально-экономических и компьютерных наук* |
| Гарифуллин Александр Михайлович  **РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ИЗУЧЕНИЯ АЛГОРИТМОВ**  *Курсовой проект*  студента образовательной программы «Программная инженерия» по направлению подготовки *09.03.04 Программная инженерия*  Руководитель  Преподаватель кафедры ИТБ  А.В. Михайлов |

Пермь, 2024 год

Аннотация

лр

Оглавление

[Введение 4](#_Toc147868660)

[Заключение 9](#_Toc147868661)

[Библиографический список 10](#_Toc147868662)

Введение

В настоящее время в Российской Федерации наблюдается рост спроса на IT специалистов, в частности, программистов [2]. Чтобы стать квалифицированным разработчикам человек должен эффективно решать поставленные задачи и с максимальными возможностями использовать языки программирования. Для развития этих навыков следует изучать алгоритмы [1].

В открытом доступе в сети «Интернет» существуют различные веб-сайты с теорией по алгоритмам: e-maxx [4], algorithmica [5], neerc.ifmo.wiki [3] и т.д. Но, для того чтобы закрепить знания, необходимо применять алгоритмы на практике. Существуют сервисы, где пользователи могут решать олимпиадные задачи на алгоритмы: Codeforces [7], Яндекс.Контест [8], acmp [6] и т.д.

После прочтения теории на каком-либо веб-сайте человек должен самостоятельно искать задачи уже на другом сервисе, т.к. на данный момент не существует решений, где будет как теория, так и сопутствующая ей практика.

В связи с вышеописанным создание веб-приложения для изучения алгоритмов, включающей в себя и теорию, и практику, является актуальной задачей.

Веб-приложение позволит не тратить время на поиск задач на алгоритмы, поможет изучить и закрепить теорию.

Объектом исследования является процесс изучения алгоритмов. Предметом исследования является веб-приложение для изучения алгоритмов.

Цель работы – разработка веб-приложение для изучения алгоритмов, включающее в себя как теорию, так и практику, позволяющую освоить теорию.

Для выполнения поставленной цели необходимо выполнить следующие задачи:

1. Провести анализ процесса изучения алгоритмов и разработать требования к системе.
2. Выполнить проектирование веб-приложения для изучения алгоритмов.
3. Выполнить реализацию веб-приложения для изучения алгоритмов.
4. Провести тестирование разработанного веб-приложения для изучения алгоритмов.
5. Провести развертывание разработанного веб-приложения для изучения алгоритмов.

Этап анализа процесса изучения алгоритмов и разработки требований к системе должен включать в себя:

1. Анализ основной документации предметной области.
2. Анализ наиболее известных существующих решений.
3. Описание сценариев использования и бизнес-процессов в нотации UML.
4. Выделение функциональных и нефункциональных требований к системе и подготовка технического задания по ГОСТ…

Этап проектирования веб-приложения для изучения алгоритмов:

1. Построение первичного отношения и его нормализация до базы данных.
2. Создание физической модели.
3. Построение и представление модели полученной базы данных в нотации UML.

Практическая значимость результатов заключается в том, что на изучение алгоритмов будет тратиться меньше времени, т.к. и теория, и практика будут в одном месте.

Веб-приложение реализуется в Microsoft Visual Studio на языке программирования C# с использованием фреймворка ASP.NET MVC. Кроме того, требуется провести компонентное тестирование приложения согласно критериям черного ящика.

**Глава 1. Анализ процесса изучения алгоритмов и разработка требований к веб-приложению для изучения алгоритмов**

В ходе анализа процесса изучения алгоритмов необходимо выполнить следующие задачи:

1. Выполнить анализ процесса изучения алгоритмов.
2. Сравнить и оценить существующие решения, которые позволяют изучать алгоритмы.
3. Сформировать список требований к веб-приложению для изучения алгоритмов.
4. Описать сценарии использования и бизнес-процессы в нотации UML.

Результатом анализа должно стать техническое задание, выполненное по ГОСТ 19.201-78.

1. Анализ процесса изучения алгоритмов.

Процесс изучения любого алгоритма можно разделить на 2 части: изучение теории и закрепление изученного на практике. С теорией можно познакомиться в специальных курсах и лекциях, книгах и статьях. Практика при изучении алгоритмов – это написание алгоритма в программе на любом языке программирования. При этом, необходимо не просто выучить алгоритм, знать его реализацию, основную идею и асимптотику, важно также научиться понимать, в каких случаях применять алгоритм, когда возникают такие случаи, и как нужно модифицировать алгоритм под конкретную задачу. Простое перепечатывание алгоритма в программу не поможет, таким способом нельзя изучить алгоритм. Для полного понимания алгоритмов существуют олимпиадные задачи по программированию, где по условиям задачи явно не сказано, как и где применить алгоритмы, какие алгоритмы вообще стоит применить в задаче. В процессе решения таких задач можно научится «видеть» алгоритмы, «придет» большее понимание алгоритмов в целом. После десятка решенных задач можно научиться применять алгоритм.

При этом алгоритмы неразрывно связаны со структурами данных, так как большинство алгоритмов используют какие-либо структуры данных в своей реализации. Также отметим, что часто алгоритмы – это не просто набор команд, приводящих к какому-то результату, это идея, которую можно использовать в различных задачах. При этом эту идею можно применять на разных уровнях: это может быть простая задача, где можно просто перепечатать алгоритм из конспекта, а может быть задача, где заметить и применить идею очень сложно.

Не стоит также забывать про то, что во время изучения алгоритмов человек может пользоваться конспектами. Особенно часто к теоретическому материалу обращаются при изучении сложных алгоритмов, где нелегко с прочтения сразу все понять. Конспекты могут быть как письменными, так и электронными. При этом следуют обращаться к наиболее достоверным и понятным ресурсам, поэтому стоит отдавать предпочтения конспектам, написанными профессионалами и экспертами в области алгоритмов и структур данных.

Важно отметить, нужно изучать разные алгоритмы одного уровня, невозможно изучить на очень высоком уровне только какой-то один вид алгоритма, при этом не зная хотя бы несколько алгоритмов среднего уровня, так как чем сложнее алгоритм, тем больше шанс того, что он будет использовать алгоритмы или идеи алгоритмов более низкого уровня. Зная несколько разных алгоритмов одного уровня, можно легче и быстрее изучить алгоритмы более высокого уровня. Это связано также с тем, что для изучения алгоритмов высокого уровня может просто не хватать умственных способностей, так как мозг будет не сильно развит и ему будет сложнее представлять идеи алгоритма. Но если человек будет изучать разные алгоритмы, то его кругозор расширится, усилится логическое и абстрактное мышление, что поспособствует тому, что изучать новые темы будет намного легче. Поэтому важно постепенно изучать разные алгоритмы одного уровня, а не углубляться сразу в какой-то один вид алгоритмов, игнорируя другие виды алгоритмов. Поэтапно с увеличением числа пройденных алгоритмов на разные темы одного уровня можно переходить к более сложным алгоритмам.

1. Анализ **наиболее известных существующих решений для изучения алгоритмов**.

Для облегчения процесса изучения алгоритмов уже существует несколько решений. Некоторые из них затрагивают только теорию, другие только практику, но есть и те, где можно встретить и теорию, и практику, но в виде исключения (т.е. в основном в системе практика, но по некоторым задачам есть и теория, или наоборот). Разделим обзор решений на две части:

1.2.1 Существующие решения с теорией по алгоритмам

В основном все решения с теорией похожи: они хранят какие-либо статьи по алгоритмам. Самыми популярными сервисами являются: Викиконспекты ИТМО [3], e-mmax[4] и Алгоритмика [5]. При этом просматривать материалы можно на каждом сайте не зарегистрировавшись.

Викиконспекты ИТМО хранят конспекты лекций по дисциплине «Алгоритмы и структуры данных» университета ИТМО. Контексты отсортированы по сложности материала, но он рассчитан для студентов, которые уже разбираются в базовых алгоритмах. Для новичков они могут показаться слишком сложными для понимания.

e-mmax хранит не только статьи по алгоритмам, но и книги по ним. Сервис относительно старый, поэтому статьи могут содержать неточности и ошибки. Также в статьях может быть представлена не самая лучшая реализация алгоритмов, так как эта реализация была написана на старых версиях языка программирования C++. В конце некоторых статей можно встретить пару ссылок на задачи на других сервисах, при этом часть ссылок уже не действительна. Отметим, что статьи распределены по темам, но нет распределения по сложности материала. При этом затрагиваются средние и выше среднего по сложности темы. Сайт рассчитан не для новичков.

Алгоритмика – относительно новый открытый проект с материалами лекций различных курсов по Computer Science, проводимых в Tinkoff Generation. Основная аудитория сайта – школьники, поэтому конспекты написаны на более простом и понятном для новичков языке. Статьи на сайте различного уровня: как для начинающих, так и для победителей международной олимпиады школьников по программированию. При этом статьи разделены по темам, но не разделены по сложности.

1.2.2 Существующие решения с практикой алгоритмов

Самым популярным и общедоступным сервисом для решения задач в Российской Федерации является Codeforces [7]. На сайте зарегистрировано более 140 000 пользователей. При этом большую часть составляют граждане других стран. Сервис предоставляет возможность решения олимпиадных задач по программированию разного уровня (от новичка до победителя международных олимпиад по программированию) на более чем 10 языках программирования и разных компиляторов. Отметим, что для каждой отправленной посылки для решения задачи можно посмотреть максимальное время выполнения программы и объем затраченной памяти в МБ. некоторых случаях можно посмотреть тесты задач. При этом есть специальный раздел обучения, где можно посмотреть лекции по некоторым темам, и пройти практику (решить задачи на эти темы). Но этот раздел посвящен не новичкам, так как там затрагивают лишь сложные темы (Дерево отрезков, система непересекающихся множеств, z-функция и другое). Дополнительно стоит отметить, что сайт предоставляешь возможность создания групп для решения наборов задач и самих наборов задач (мэшапов). Группы для решений могут быть как общедоступными (любой зарегистрированный пользователь может вступить в группу и начать решать задачи), работающие по приглашению пользователей (только приглашенные пользователи могут решать задачи), по подтверждению (любой пользователь может подать заявку на вступление в группу и начать решать задачи, после одобрения заявки). При этом в мэшапах можно использовать уже существующие задачи из общедоступного архива задач (архив содержит около 10 000 задач), задачи из других отрытых мэшапов (около 2000 задач), а также можно самому создать задачи и добавить их в мэшап (для этого стоит воспользоваться системой Polygon.Codeforces) [9]. Дополнительно отметим, что к каждому мэшапу можно прикрепить материалы (ссылка на что-либо, либо файл формата pdf). Помимо возможности хранения мэшапов для решения задач, можно легко отслеживать процесс решения участниками как каждого мэшапа по отдельности, так и всех мэшапов в общем, так так в каждом мэшапе есть свое «положение», где будут указаны все пользователи, решавшие задачи из этого мэшапа, а также можно легко добавить рейтинг пользователей, где перечислить мэшапы для составления рейтинга, после этого можно будет смотреть динамику решения задач по всем мэшапам.

Стоит также отметить Яндекст.Контест [8] и acmp [6], где собрано множество олимпиадных задач, но данные сервисы предоставляют куда меньшие возможности, чем Codeforces[7]. Также отмечу, что просматривать задачи можно в каждом сервисе не зарегистрировавшись, но отправлять посылки на задачи могут лишь зарегистрированные пользователи.

1.2.3 Итоговый обзор существующих решений

Лучшим сервисом с теорией для изучения алгоритмов считаю Алгоритмику [5], так как конспекты сайта относительно других сервисов самые новые, при этом написаны на более простом и понятном язык, и уже использовались в кружках Tinkoff Generation для обучения школьников алгоритмам. Но на сайте нет определенного плана изучения конспектов, поэтому самостоятельное изучение алгоритмов будет затруднительно. Также на сайте нет возможности как-либо испытать знания на практике.

Лучшим сервисом для практики считаю Codeforces, так как он предоставляет большой набор задач для решения, а также возможности для организации тренировок (группы с мэшапами). Но на сайте нет возможности новичку изучить алгоритмы. При этом задачи на одну тему и одной сложности, согласно сервису, могут сильно отличаться по реальной сложности, так как сайт существует уже давно (более 10 лет) и старые задачи заметно проще новых, но при этом рейтинг старых задач не меняется под новые реалии.

Соединив теорию с Алгоритмики и задачи с Codeforces, можно получить сервис, который будет одновременно предоставлять лучшую теорию и лучшую практику. Разрабатываемая система должна будет избавиться от минусов уже существующих решений, поэтому стоит сделать следующее:

1. Конспекты должны быть по сложности на уровне Алгоритмики.
2. Должно быть разделение конспектов по темам.
3. Должно быть разделение конспектов по сложности.
4. Стоит добавить пошаговый план изучения алгоритмов.
5. Должна быть возможность решать задачи, при этом задачи должны быть отсортированы по реальной сложности.

Стоит отметить, что в одиночку реализовать возможности системы для безопасного решения задач хотя бы на самых популярных языках программирования, а также для добавления задач в тренировки невероятно трудно и занимает огромное время, из-за чего работа займет намного больше времени, чем уделено на курсовую работу 3 курса, поэтому вместе с руководителем курсовой работ было принято решение: не разрабатывать такие возможности, а использовать уже готовое, а именно на Codeforces будет создана группа с наборами задач на определенные темы, при этом в разрабатываемой системе в конспектах должны быть ссылки на эти наборы задач. В таком случае пользователь сможет прочитать конспект, после чего перейти по ссылке из этого конспекта на задачу на сайте Codeforces и решить ее. При этом пользователь не тратит время на поиски задач для закрепления теории, так что практическая значимость соблюдена. В дальнейшем можно добавить возможность решать задачи прямо в разрабатываемой системе, или добавить какой-либо анализ информации из группы с Codeforces.

Также стоит разделить пользователей на три вида: редакторы, читатели и администраторы. Читатели могут читать конспекты, проходить план изучения алгоритмов. Им не нужно регистрироваться и входить в систему. Редакторы могут добавлять, редактировать, удалять и читать конспекты, редактировать план обучения. Для работы они должны регистрироваться и авторизовываться в системе. При этом редакторы не могут сами зарегистрироваться, им выдают логин и пароль. Администраторы могут делать все, что могут делать редакторы, при этом они создают новых редакторов и администраторов.

1. Анализ требований к системе для изучения алгоритмов.

Определим функциональные (определяют задачи, которые должны выполняться) и нефункциональные (свойства и ограничения, которые должны быть) требования к системе.

Список функциональных требований к системе:

1. Пользователи должны иметь возможность войти в систему через уникальный логин (имя пользователя) и пароль.
2. В системе должны быть конспекты алгоритмов, представленные в виде статей.
3. В системе должен быть пошаговый план изучения алгоритмов, который состоит из упорядоченного списка конспектов.
4. Редакторы должны иметь возможность добавлять новые конспекты.
5. Редакторы должны иметь возможность редактировать существующие конспекты.
6. Редакторы должны иметь возможность удалять существующие конспекты.
7. Редакторы должны иметь возможность просматривать существующие конспекты.
8. Читатели должны иметь возможность просматривать существующие конспекты.
9. Конспекты должны быть распределены по темам.
10. Редакторы должны иметь возможность задавать конспектам темы.
11. Редакторы должны иметь возможность создавать темы для конспектов.
12. Редакторы должны иметь возможность удалять темы для конспектов.
13. Редакторы должны иметь возможность редактировать темы для конспектов.
14. Редакторы должны иметь возможность просматривать темы для конспектов.
15. Редакторы должны иметь возможность добавлять конспекты в пошаговый план для изучения алгоритмов.
16. Редакторы должны иметь возможность удалять конспекты из пошагового плана для изучения алгоритмов.
17. Редакторы должны иметь возможность изменять порядок конспектов в пошаговом плане для изучения алгоритмов.
18. Редакторы должны иметь возможность просматривать конспекты в пошаговом плане для изучения алгоритмов.
19. Читатели должны иметь возможность просматривать конспекты в пошаговом плане для изучения алгоритмов.
20. Администраторы должны иметь возможность создавать новых пользователей в системе.
21. Администраторы должны иметь возможность редактировать пользователей в системе.
22. Администраторы должны иметь возможность удалять пользователей в системе.
23. Администраторы должны иметь возможность просматривать пользователей в системе.
24. Администраторы должны иметь возможность выдавать роли пользователям.
25. Администраторы должны иметь возможность изменять роли пользователей.
26. Администраторы должны иметь доступ ко всем функциями системы: функции редакторов и свои дополнительные привилегии.

Список нефункциональных требований к системе:

1. База данных системы должна быть нормализована как минимум до третьей нормальной формы.
2. Пароли пользователей должны храниться в зашифрованном виде.
3. Время загрузки конспектов должно быть минимальным для обеспечения удобства пользователей.
4. Запросы на добавление, редактирование и удаление должны обрабатываться быстро и эффективно.
5. Интерфейс должен быть интуитивно понятным и удобным в использовании.
6. Редакторы и администраторы обязаны выполнить вход в систему через логин и пароль для доступа к своим правам.
7. Читатели не должны входить в систему для просмотра контента.
8. Сценарии использования и бизнес-процессы.

вы

1. Результаты анализа процесса изучения алгоритмов.

вы

Заключение

опр

Библиографический список

1. <https://habr.com/ru/articles/279453/>
2. <https://www.cnews.ru/news/top/2023-07-04_v_rossii_vzryvnoj_rost_sprosa>
3. <https://neerc.ifmo.ru/wiki/index.php?title=Заглавная_страница>
4. <https://e-maxx.ru/algo/>
5. <https://ru.algorithmica.org/>
6. <https://acmp.ru/>
7. <https://codeforces.com/>
8. <https://contest.yandex.ru/>
9. https://polygon.codeforces.com/