**XXIX Открытая московская инженерная конференция школьников «ПОТЕНЦИАЛ» на площадке НИУ «МЭИ»**

Секция: информационные технологии

«Разработка ПО»

**«Словарь программиста»**

Либерова Олеся Георгиевна,

ученица 9 “В” класса ГБОУ школы № 1080

Руководитель Е. Ю. Киселева,

Учитель информатики ГБОУ школы № 1080

Москва, 2020 г.

Введение

В современном мире программирование — пользующаяся спросом деятельность. Чтобы стать профессионалом в этой области, надо много и регулярно работать, а также необходимо владеть специальным запасом слов.

Меньше чем за минуту google выдает чуть менее 4 миллионов ответов на одноименный с названием проектный работы поисковый запрос. В сравнении со страницами, открывающимися в поиске браузера, моя программа имеет значительную изюминку — тестовую версию понятий, изученных мною в среде профессиональных наставников, а не просто словарь понятий с общепринятым определением, как к примеру известные книжные пособия англо-русских толковых словарей. Наравне с обучающими сайтами выводятся и платные курсы, в которых, как вам обещают, предоставят практичные знания, на деле же деньги на ветер и никакого продолжения после теории. А как известно, читать и знать мало, важно понимать и делать.

Одна из проблем современного российского образования заключается в отсутствии побуждения к действию для того, чтобы школьники лучше понимали необходимость определенного материала, предмета или темы. Часто школьники забрасывают идею изучать программирование по причине большого количества непонятных слов.

Цель

Создать электронный ресурс для улучшения знаний терминологии по программированию

Задачи

1. Проанализировать характерные особенности программистов на примере знакомых, изучающих программирование.
2. Придумать организованное запоминание терминов.
3. Создать систему – словарь.
4. Реализовать авторизацию пользователя в системе.
5. Создать открытие формы при нажатии на кнопку.
6. Поставить перед пользователем вопросы с вариантами ответов.
7. Разделить уровни подготовки.
8. Реализовать возможность добавления в словарь понятий и определений.
9. Систематизировать и обобщить результаты, сформулировать выводы.
10. Написать статью с описанием хода работы.
11. Представить проект научной общественности.

## Технологические решения

В программном коде задействованы базы данных, библиотеки-модули языка python и c++, основы построения сайтов на django, связка html с python-скриптом, функции, осуществляющие смену пользователя и языка (английский или русский).

Словарь программиста.

## Актуальность

В отличие от отечественной системы образования шведская система образования много внимания уделяет практичности получаемых знаний. Школа старается давать информацию разными способами: учебники, обучающие статьи и приложения, дискуссии-дебаты и записи-конспекты. Таким образом обеспечивается более глубокое понимание материала. Часто бумажные носители теряются, могут промокнуть или не эстетично порваться. В такие моменты было бы удобно иметь доступ к информации под рукой, то есть в интернете, где жители человеческих муравейников проводят большую часть времени в течение дня.

Отвергаешь - предлагай, исходя из этого и появилась идея создать электронный ресурс для программистов, включающий в себя возможности записной книжки: конспектирование трудоёмких слов, истин и словосочетаний в удобной для пользователя форме, тестом с отработкой указанного материала.

Для такого электронного ресурса важно: лаконичность интерфейса, рабочий скрипт, направленный на принятие полученной информации от пользователя и её последующей обработки, но главное – оригинальность исполнения. Написанное приложение познакомит с причудливостью сленговых сокращений программистов и заимствованными словами из других языков.

## Ход работы

Было задумано разработать независимое приложение в связи с защитой проекта по теме «Qt дизайнер» в «Яндекс.Лицее». Документации, приложенные к урокам, помогли мне справиться с базовыми трудностями, вроде грамотного написания sql-запросов, закрытие программы при нажатии на «крестик» или обработка события при нажатии на одну из кнопок, соответственно представляющих множественный выбор.

Создание визуальной части приложения велось с самого начала. Самыми сложными в написании рабочего кода оказались алгоритм авторизации пользователя, установка прозрачности на объекты; сопоставление ответа, полученного от пользователя с указанными в базе данных.

С виду вход в личный кабинет на сайте выполнить несложно, но на самом деле нужно отписать возможные варианты неправильного ввода данных: заглавный или прописной символ, имеется ли такой логин уже и выпадение нескольких текстов о допущенной ошибке подряд.

Чтобы интерфейс программы был понятен и удобен пользователю в работе, было установлено правильное месторасположение относительно композиции: как сделать окно визуально незасоренным, но с множеством функций?

Следующим этапом проекта стало создание веб-сервера, перенос приложения на разработанную платформу для обеспечения свободного доступа.

В процессе работы был произведен выбор технологии для реализации проекта. Рассматривались различные версии от чистого html до комбинации различных языков, таких как java и php. В результате анализа был выбран более удобный для разработки проекта – интерпретатор django, позволяющий значительно сократить написание вторичных процедур и функций, используя встроенные. К примеру, механизмы авторизации, доступа к базам данных. Для переноса приложения в django необходимо было написать все с нуля. В рамках работы были освоенны методы построения веб-сервера на технологии django, а именно – механизм создания url`ов и view`ек, работа с шаблонами, технологии css. Язык таблиц стилей позволяет задать необходимое визуальное оформление путем присваивания конкретных атрибутов, таких как цвет, шрифт, позиционирорвание объекта отдельным классом.

За шестью часами, проведенного за просмотром вебинара по основам django, были переписаны базы данных с вопросами и личными данными каждого пользователя, автоматическое сохранение последнего вопроса в личном кабинете.

На той же неделе было пару мозговых штурмов, в результате чего были настроены счетчики с передачей данных, отсылки на классы Вход и Выход из аккаунтов, исправлены привязки объектов с интерфейсом, откоректированно принятие ответов на втором уровне сложности через прописные буквы, сменены размеры заголовков, настроено отображение списка всех вопросов только после прохождения всего теста.

Подводя итог, я прогнала теорию из учебников через себя, составила вопросы с возможными вариантами ответов, некие уловки в темах, где чаще всего встречаются провалы памяти. Все теоретические материалы были собраны с языка программирования python.

В ходе работы над проектом был сделан один существенный вывод: «Хочешь что-то понять – сделай это самостоятельно до такой степени, чтобы внятно объяснить материал коллеге». Так ты сможешь не только повторить полученные знания, но и компактно систематизировать.

Заключение

Проект может пригодиться специалистам в области информационных технологий в связи с помощью в конкретной задаче – усвоение полученного материала и минимизации потраченного времени на заучивание.

Реализованы методы идентификации как в начале на версии независимого приложения, так впоследствии на механизмах встроенной авторизации. Созданы различные формы как в локальной версии, так и в приложении открытого доступа.

В ходе работы были обнаружены дальнейшие перспективы развития веб-сервера: возможность изменения вариантов ответов в создании теста, энциклопедическая библиотека с возможностью поиска по ней слов и явлений, рассылки-напоминания о тренировке на указанную пользователем почту в личном кабинете.

Использованная литература:

<https://expert.ru/countries/2012/13/bez-lishnej-strogosti/>

<https://pythonworld.ru/gui/pyqt5-eventssignals.html>

<http://htmlbook.ru/samhtml/spiski/markirovannyy-spisok>

<https://django-grappelli.readthedocs.io/en/latest/quickstart.html#installation>

<https://lyceum.yandex.ru/courses/166/groups/1379/lessons/1257>

<https://pandia.ru/text/77/275/19660.php>

<https://itproger.com/course/django/6>