Заключение. Разработан офлайн-компилятор языка Pascal для устройств на операционной системе Android, которые полностью поддерживает все возможности компилятора языка, как и на персональном компьютере, а также множество дополнительных библиотек для Android-устройств.

Список цитируемых источников

1. Android Studio [Электронный ресурс]. — Режим доступа: http://wnfx.ru/android-studio-ide-ot-google/. — Дата доступа: 18.02.2019.

УДК 004.421

А. А. Ермакова
учреждение образования «Барановичский государственный университет», Барановичи

ВЕБ-ПРИЛОЖЕНИЕ «ЕЖЕДНЕВНИК»

Введение. В современном быстроразвивающемся мире повышается занятость человека: множество обязанностей, целей и задач, которые, зачастую, нужно реализовать в течение всего лишь одного дня. Венцом успеха будет являться правильное распределение времени.

Основная часть. В правильном распределении времени помогает ежедневник: человек грамотно расписывает свой день буквально по минутам и, как итог, успевает всегда и везде. На данный момент актуальность использования web-приложения «Ежедневник» очень высока, так как ежедневник помогает не упускать из виду ни одной детали, человек может анализировать эффективность своей работы и даже находить ошибки в своих поступках или действиях.

Целью работы является разработка web-приложения «Ежедневник».

Достижение поставленной цели требует решения следующих задач: разработка методов и моделей представления ежедневника; разработка информационной модели подсистемы (структуры базы данных) учета пользователей и наполнение разработанной БД соответствующей информацией о записях; тестирование программы с использованием разработанной БД; описание алгоритмов программных модулей; описание полученных результатов.

Веб-приложение — клиент-серверное приложение, в котором клиент взаимодействует с сервером при помощи браузера, а за сервер отвечает веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными службами [1].

Клиент-сервер — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг (сервисов), называемыми серверами, и заказчиками услуг, называемыми клиентами.

Сервер — это программа, представляющая какие-то услуги другим программам и обслуживающая запросы клиентов на получение ресурсов определенного вида. Клиент — это программа, использующая услугу, представляемую программой сервера.

Часто люди клиентом или сервером просто называют компьютер, на котором работает какая-либо из этих программа. В сущности, клиент и сервер — это роли, исполняемые программами. Клиенты и сервера физически могут находится на одном компьютере. Одна и та же программа может быть и клиентом, и сервером одновременно [2].

Java-WEB приложение представляет собой систему, которая реализует структуру MVC(Model-view-controller). В данной структуре за отображение информации отвечает «view» а за реакции на действия пользователя «controller».

Приложение состоит из двух составляющих:

— сервера (в данной работе используется Tomcat 9);

— клиента (может использоваться любой web-браузер, в данном проекте использован Google Chrome).

Окно клиента представлено на рисунке 1.

Для того, чтобы войти в аккаунт либо зарегистрировать новый, необходимо пройти на страницу «авторизация/регистрация». Страница с формами «авторизация/регистрация» представлена на рисунке 2.

Личная страница пользователя представлена на рисунке 3.

Рисунок 1 — Представление клиента

Рисунок 2 — Представление авторизации/регистрации

Рисунок 3 — Представление личной страницы

Работа программы начинается с главной страницы, где пользователь может просмотреть информацию о сайте, зарегистрироваться либо войти на свою личную страницу, введя имя и пароль. При неуспешной авторизации пользователя перенаправит на главную страницу. После успешной авторизации, перейдя на свою личную страницу пользователь может выбрать дату и записать какую-либо информацию. При выборе даты проверяется существует ли заметка пользователя с выбранной датой и, если она существует, то она будет выведена на экран для изменения или ознакомления. При изменении записи происходит обновление данных для текущей заметки в базе данных.

Заключение. Разработанное приложение реализовано с использованием языка программирования Java в среде Intellj IDEA, с использованием MySQL и языка структурированных запросов SQL. Применение базы данных в качестве хранилища информации позволяет оптимально и эффективно хранить информацию, ее структурировать.

Список цитируемых источников

1. ВикипедиЯ. Свободная энциклопедия. Веб-приложение [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Веб-приложение (дата доступа: 14.12.2018).
2. Основные понятия архитектуры клиент-сервер [Электронный ресурс]. URL: http://bukvi.ru/computer/osnovnye-ponyatiya-arxitektury-klient-server.html (дата доступа: 14.12.2018).

УДК 004.021

С. Ю. Мальчиков
Учреждение образования «Барановичский государственный университет», Барановичи

РАЗРАБОТКА ПРОЕКТА РЕШЕНИЯ ЗАДАЧИ КОММИВОЯЖЁРА

Введение. Оптимизация производства, в широком смысле слова, находит применение в науке, технике и в любой другой области человеческой деятельности. Оптимизация - целенаправленная деятельность, заключающаяся в получении наилучших результатов при соответствующих условиях. Поиски оптимальных решений привели к созданию специальных математических методов, и уже в 18 веке были заложены математические основы оптимизации (вариационное исчисление, численные методы и др). Однако, до второй половины 20 века, методы оптимизации во многих областях науки и техники применялись очень редко, поскольку практическое использование математических методов оптимизации требовало огромной вычислительной работы, которую без ЭВМ реализовать было крайне трудно, а в ряде случаев - невозможно. При наличии ЭВМ решение задачи заметно упрощается [1].

Целью данного проекта является разработка приложения для решения задач оптимизации методом ветвей и границ (алгоритм Литтла) и генетическим алгоритмом.

Задачи проекта: решить задачу метод ветвей и границ (алгоритм Литтла); решить задачу генетическим алгоритмом; сравнить результаты; реализовать данные методы в приложении.

Актуальность выбранной темы заключается в следующем. Для решения оптимизационных задач нередко прибегают к формулировке поставленной задачи в виде каких-то хорошо известных математических задач: транспортная задача, задача поиска оптимального пути (задача коммивояжера) и другие. Сформулированную таким образом задачу решают, используя такие математические методы, как метод ветвей и границ, симплексный метод, метод Фогеля (приближенного решения), генетический алгоритм и другие [2,3].

Основная часть. В данном проекте используется метод ветвей и границ (алгоритм Литтла), так как является универсальным методом решения задачи дискретного программирования.

Алгоритм Литтла применяют для поиска решения задачи коммивояжера в виде гамильтонова контура. Данный алгоритм используется для поиска оптимального гамильтонова контура в графе, имеющем N вершин, причем каждая вершина i связана с любой другой вершиной j двунаправленной дугой. Каждой дуге приписан вес C_{ij}, причем веса дуг строго положительны ($C_{ij} \geq 0$). Веса дуг образуют матрицу стоимости. Все элементы по диагонали матрицы приравнивают к бесконечности ($C_{ij} = \infty$). В случае, если пара вершин i и j не связана между собой (граф не полносвязный), то соответствующему элементу матрицы стоимости приписываем вес, равный длине минимального пути между вершинами i и j. Если в итоге дуга (i и j) войдет в результирующий контур, то ее необходимо заменить соответствующим ей путем [4].

В качестве второго метода решения был выбран генетический алгоритм. Суть подхода в том, что прежде генерируется множество маршрутов. Это множество называется популяцией.

Эффективность применения ГА к задаче коммивояжёра зависит также и от настройки операторов мутации и кроссинговера.

Мутация, как незначительное (точечное) случайное изменение в хромосоме может осуществляться несколькими способами. Наиболее распространенным является выбор двух случайных ген и их обмен местами. Цель оператора мутации — расширение пространства поиска за счет случайных точечных изменений в хромосоме. В классическом варианте действие оператора кроссинговера состоит в следующем: выбираются две определенные хромосомы (случайным образом), далее также случайным образом определяется точка кроссинговера и происходит обмен участками хромосом.

Однако полученные хромосомы не всегда являются допустимыми маршрутами коммивояжера (как в данном примере). Возможный вариант осуществления кроссинговера — брать не две хромосомы, а одну. В этом случае действие оператора кроссинговера будет состоять в повороте участка хромосомы либо в его циклической перестановке. В проекте был выбран двухточечный кроссинговер.

В случае задачи коммивояжёра основные понятия генетического алгоритма имеют следующий смысл: ген — это номер вершины графа городов (номер города), через который проходит маршрут коммивояжера; хромосома — последовательность ген, определяющих замкнутый маршрут, проходящий через все заданные города (вершины графа); популяция — набор из нескольких возможных маршрутов. Состояние популяции — множество возможных маршрутов, удовлетворяющих условиям задачи. Функционал качества — длина маршрута [5].