ГБОУ “Школа 444”

Двойных Артём

Забродин Илья

Иванов Георгий

10 класс В

Уменьшение веса портфеля ученика

с помощью приложения

Руководитель проектной деятельности:

Горшкова Анастасия Игоревна

Москва 2021

Оглавление

[Введение 3](#_Toc72865283)

[Цель 3](#_Toc72865284)

[Задачи 3](#_Toc72865285)

[Актуальность 3](#_Toc72865286)

[Используемые технологии 4](#_Toc72865287)

[Структура программного продукта 5](#_Toc72865288)

[Взаимодействие программного кода на C++ и Python 6](#_Toc72865289)

[Сортировка данных (равномерное распределение веса учебников и рассадка учеников) 6](#_Toc72865290)

[Приложение 7](#_Toc72865291)

[База данных 9](#_Toc72865292)

[Заключение 10](#_Toc72865293)

# Введение

Все школьники России ежедневно сталкиваются с проблемой тяжёлых школьных портфелей. Объём печатной литературы, которую должен брать с собой на занятие школьник весьма значительный. По существующим правилам все ученики в классе на занятия должны приносить учебники. Таким образом на каждом уроке на одной парте находится два одинаковых учебника (в случаях, если ученики сидят за партами по двое). Таким образом, если бы соседи по парте носили один учебник на двоих, средний вес портфеля ученика уменьшился бы как минимум в два раза.

**Цель**

Создать приложение, помогающее распределять учебники между учениками, сидящими за одной партой. Также приложение определяет рассадку учеников на каждом из уроков.

# Задачи

В своём проекте были поставлены следующие цели:

1) Разработка алгоритма равномерного распределения веса учебников между учениками класса с формированием списка рассадки учеников в классе.

2) Разработка приложения под ОС Windows и Android для осуществления удобства использования ПО.

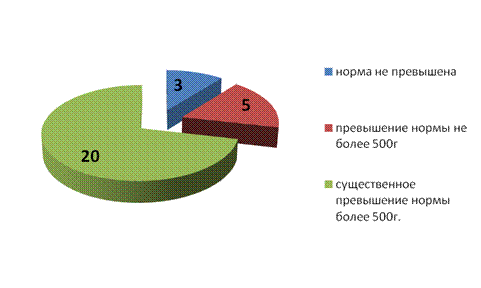
# Актуальность

По данным ***Федеральной службы по надзору в сфере защиты прав потребителей и благополучия человека*** *(*Роспотребнадзор) в 3 и 4 классе вес портфеля с учебниками не должен превышать двух килограммов. Пятиклассникам и шестиклассникам рекомендован вес 2,5 килограмма. В 7-8 – классах не больше 3,5 килограммов. В 9-11 классах максимальный вес - 4 килограмма. При этом сам ранец в старших классах не должен весить больше килограмма, а в младшей школе -700 граммов (сообщение ТАСС от 19 августа 2019 г.)

Также Роспотребнадзор заявляет, что эти нормы превышаются в два-три раза, так как портфель с учебниками на пять уроков весит по меньшей мере 5 килограмм. Наше приложение поможет снизить этот вес в 2 раза.

Анализ, рынка специализированного и общего программного обеспечения, показал, что в настоящее время ПО, решающие данную задачу отсутствуют.

Диаграмма превышения допустимого веса портфеля ученика:



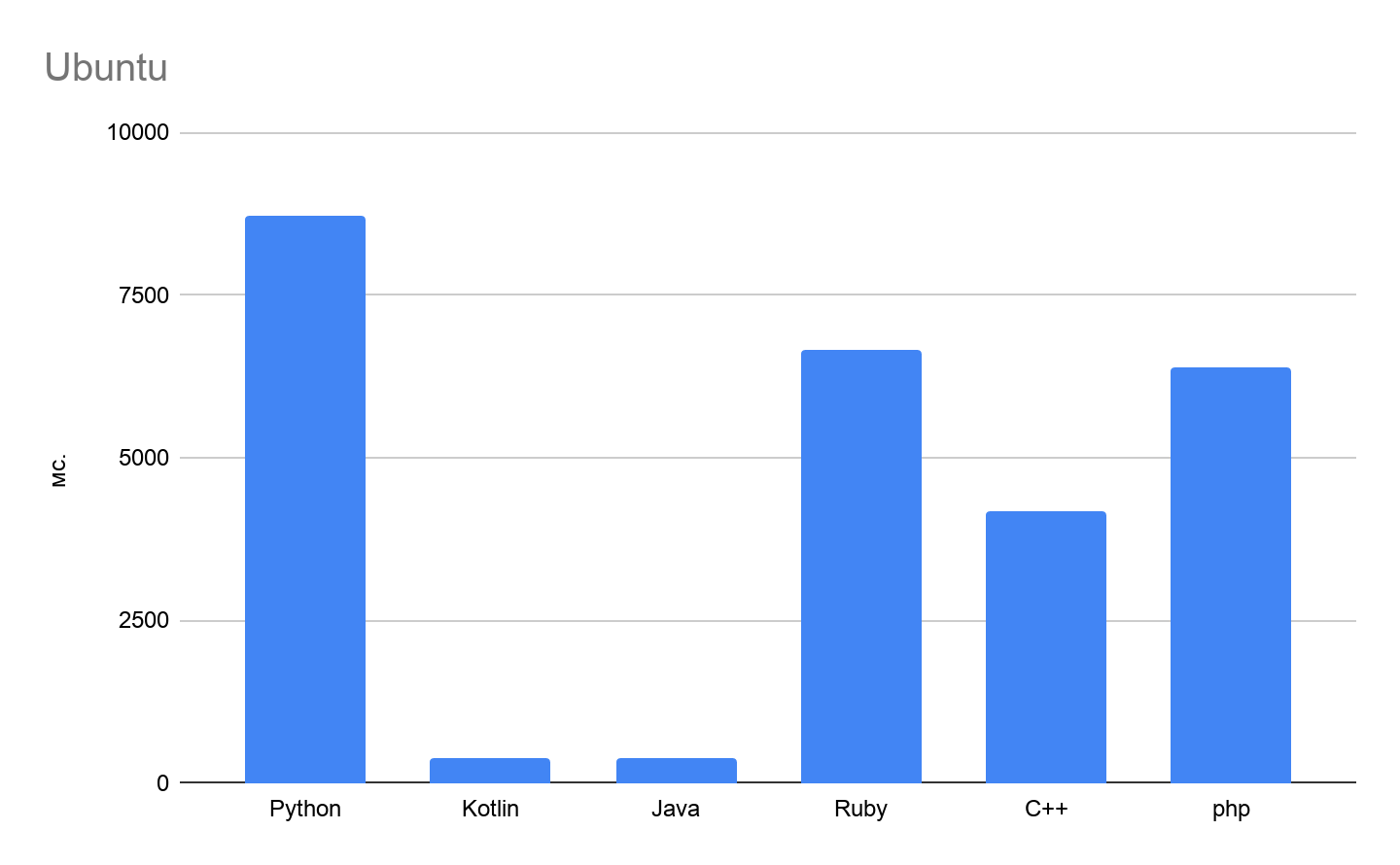
# Используемые технологии

Для решения поставленной задачи было принято решение использовать в качестве базовых технологий современные программные средства. Основными критериями при выборе программных средств были:

1. Наличие достаточного количества справочных материалов и профессиональных форумов по программированию
2. В полной мере применение современных технологий
3. Возможность построения аппаратно-независимого пользовательского интерфейса
4. Скорость обработки при выполнении задач по сортировке
5. Масштабируемость решения.

Всем этим критериям удовлетворяли такие языки программирования, как Python и C++. В качестве базы данных использовалась база sqlite3. Использование Python при разработке приложения (пользовательской части), позволило задействовать весьма широкий функционал, а также в полной мере использовать совместимость со множеством операционных систем.

В качестве языка программирования для обработки данных мы использовали язык С++, учитывая возможности построения наиболее оптимальных с позиции скорости выполнения алгоритмов сортировки.



\* время выполнения алгоритма *(“Сравнение скорости работы языков программирования на примере решения задачи обучения нейронной сети”, habr.com; karpach2000 18 апреля 2020 в 00:22*)

Для подключения к базе данных мы использовали язык SQL, поскольку данная технология позволяет без дополнительных сложных преобразований использовать различные серверные платформы хранения данных.

# Структура программного продукта

Итоговый программный продукт представляет из себя “симбиоз” написанной на Phyton и С++. Основания такого построения были описаны ваше. На первом этапе создания программного продукта было принято решения не интегрировать программный код, написанный на С++ в программный код Phyton. Обмен данными между двумя модулями осуществлялся через внешний файл формата .txt следующей структуры:

В программном модуле Phyton использовался механизм запуска внешнего приложения. А программный код с++ был скомпилирован в виде исполняемой программы под ОС Windows (на данном этапе).

# Взаимодействие программного кода на C++ и Python

Для взаимодействия программного кода на C++ и Python мы использовали обычный txt файл (код С++ читает данные из файла и возвращает в файл обработанные данные). Мы выбрали именно txt файл для наглядности и простоты соединения. Также, в дальнейшем его можно легко заменить на прямое соединение между языками.

|  |
| --- |
|  |

# Сортировка данных (равномерное распределение веса учебников и рассадка учеников)

Обработку данных и выполнение сортировки реализовано на языке С++.

В качестве входных данных программа использует имена учеников, вес учебника для каждого из уроков и расписание уроков.

Результатом выполнения программы является имя соседа для конкретного урока и список учебников для каждого ученика.

Для реализации этого функционала мы поддерживаем отсортированный массив, в котором лежат веса портфелей учеников. При добавлении очередного урока, ученики разбиваются на пары с помощью “жадного алгоритма”, что позволяет максимально уравнять вес портфелей всех учеников. Это позволяет минимизировать максимальный вес портфеля.

Изображение выглядит как текст

Автоматически созданное описание

# Приложение, интерфейс пользователя

Создание приложения мы выполнили на языке Python.

Изображение выглядит как текст

Автоматически созданное описание

Приложение обеспечивает удобное взаимодействие пользователя с базой данных и программой обработки данных.

Для распределения прав доступа в приложении было выбрано 3 модели пользователя приложения: ***Администратор***, ***Учитель*** или ***Ученик***

|  |  |
| --- | --- |
| Модель (профиль) | Описание |
| ***Администратор*** | Максимальная степень доступа к данным. Добавление, удаление пользователей. Запрет/разрешение авторизации пользователя. Изменение учётных данных пользователя. |
| ***Учитель*** | Формирование расписания, внесение данных по отсутствующим в классе, корректировка списка рассадки учеников |
| ***Ученик*** | Просмотр расписания и дневного списка учебников |

# Главная страница приложения

На главной странице пользователь может авторизоваться как Администратор, Учитель или Ученик.

|  |
| --- |
| Изображение выглядит как текст  Автоматически созданное описание |

Далее, если вы зарегистрировались как Администратор, вы можете редактировать таблицу учеников и делать выборки по ней.

|  |
| --- |
|  |

Если вы зарегистрировались как ***Ученик***, то вы можете узнать набор учебников, который вам необходимо взять на следующий день, а также своего соседа для каждого из уроков.

# 

# Структура База данных

**Изображение выглядит как текст, квитанция

Автоматически созданное описание**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Time\_table*** | | | |
| Наименование поля | Тип данных | Длина | Описание |
| Id\_lesson |  |  |  |
| number\_lesson |  |  |  |
| name\_lesson |  |  |  |
| Id\_teacher |  |  |  |
| day |  |  |  |
| id\_book |  |  |  |
| class\_number |  |  |  |
| class\_char |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Students*** | | | |
| Наименование поля | Тип данных | Длина | Описание |
| id\_users |  |  |  |
| surname |  |  |  |
| name |  |  |  |
| second\_name |  |  |  |
| school |  |  |  |
| class\_number |  |  |  |
| class\_char |  |  |  |
| login |  |  |  |
| password |  |  |  |
| main\_admin |  |  |  |
| admin |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ***teachers*** | | | |
| Наименование поля | Тип данных | Длина | Описание |
| id\_teachers | String | 256 |  |
| surname |  |  |  |
| name |  |  |  |
| second\_name |  |  |  |
| age |  |  |  |
| subject |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Students\_book*** | | | |
| Наименование поля | Тип данных | Длина | Описание |
| id\_book | String | 256 |  |
| subject |  |  |  |
| weight |  |  |  |
| author |  |  |  |
| class\_number |  |  |  |

# Заключение

В рамках данного проекта нам удалось решить поставленные задачи. Заложенные на данном этапе принципы и используемые технологии позволяют говорить о возможности построения полноценной аппаратно-независимой системы, которая позволит существенно (в прямом смысле) облегчить жизнь школьника.

Одновременно сейчас мы понимаем возможные пути усовершенствования данного программного продукта, а так же видим возможные пути расширения функциональных возможностей.

В качестве возможных шагов для расширения функционала данного программного продукта мы видим:

1. Возможность настройки пользовательского интерфейса (цветовая палитра, вид сообщений, способ представления информации на экране)
2. Интеграция программного кода с++ в код Phyton
3. Разработка приложения под ОС Android и MacOs
4. Отладка метода хранения данных, с позиции контроля целостности и защиты от утечки данных третьим лицам
5. Тестовая эксплуатация программного продукта в рамках одного класса, одной школы …

# Источники

* Исследование Роспотребнадзора - <https://rg.ru/2019/10/12/reg-urfo/rospotrebnadzor-rancy-shkolnikov-vesiat-vdvoe-vyshe-normy.html>
* Учебник Python - <https://rg.ru/2019/10/12/reg-urfo/rospotrebnadzor-rancy-shkolnikov-vesiat-vdvoe-vyshe-normy.html>
* Учебник C++ - https://ravesli.com/uroki-cpp/