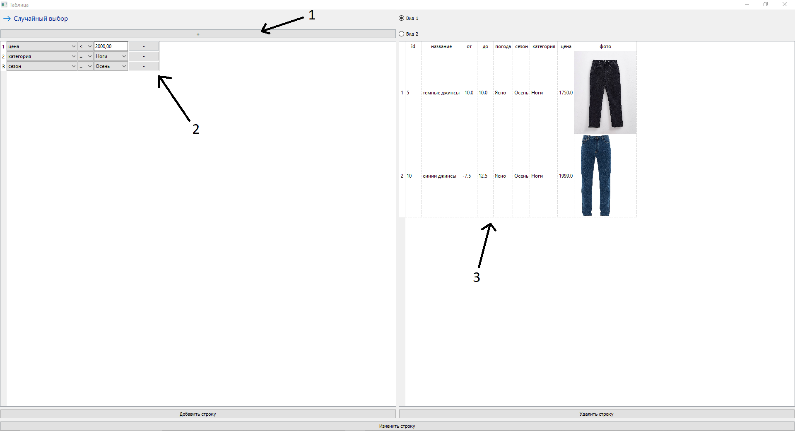
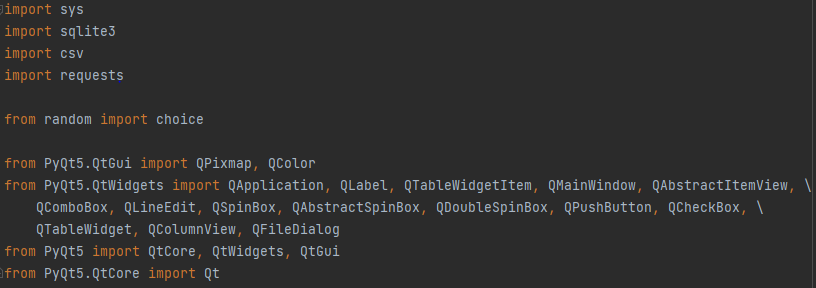
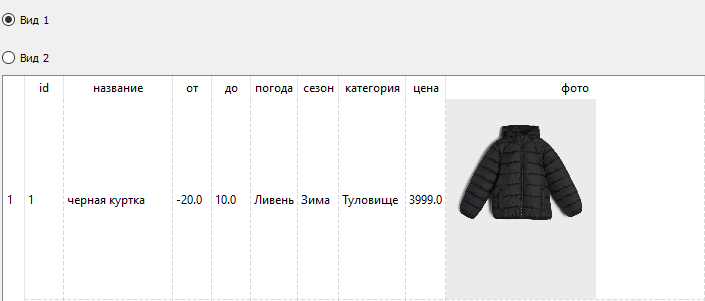
**Отчет по Qt проекту “Каталог одежды”**

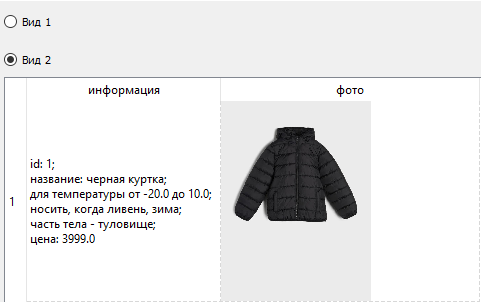
Проект сделан учеником 8”А” класса школы №509

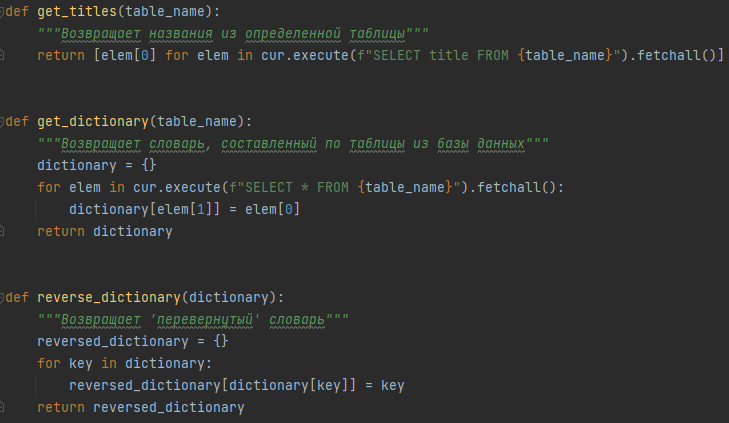
Хорошавиным Арсением

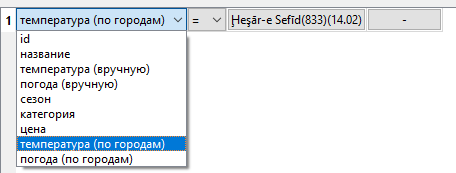
Лично у меня никогда не было проблем с выбором одежды, но я воссоздал именно такую идею, потому что посчитал ее лучшей. А потому получается, что мой проект является каталогом одежды с возможностью искать наряд по заданным характеристикам или условиям, для упрощения выбора одежды. Правда, я так и не определился - является ли мой проект каталогом домашней одежды или каталогом одежды для покупки, впрочем на его реализацию это не повлияло.

Так что же насчет реализации? А вот что: используются 6 библиотек: sys, sqlite3, csv, random, requests и, конечно же, PyQt5. Самый объемный класс по коду – это FindWidget, по названию уже можно догадаться о его функции – он ищет одежду по заданным условиям. Этот класс включает в себя кнопку для добавления условий (1), таблицу, отображающую сами условия, в которой их можно менять (2), а также таблицу с найденной одеждой (3). Основные методы следующие: find\_items, add\_if, change\_if, delete\_if, по названиям вполне можно догадаться, что они делают.

 Также есть класс ShowData – его окно, как раз-таки и изображено на рисунке со стрелками. Кроме того, что он может удалять, менять и добавлять строки с характеристиками одежды (это методы change\_line, delete\_line, add\_line) и того, что он может менять вид таблицы, отображающую одежду (change\_look), в нем ничего особого.

 Следующий класс – это RandomClothes. На его окне располагается 4 FindWidget’а, каждый из которых отображает только один предмет гардероба, а не несколько, также они отличаются тем, что у них есть изначальное обязательное условие – категория одежды, и только по этой категории они смогут выводить одежду.

 Еще можно найти 3 интересные функции в моем коде: get\_titles, get\_dictionary, reverse\_dictionary. Тут все просто: get\_titles возвращает названия чего-либо, что привязано к какому-то идентификатору, в моем случае это категория одежды, сезон и погода; get\_dictionary работает почти также, как и get\_titles, но возвращает словарь с видом {название: идентификатор}; reverse\_dictionary возвращает ‘перевернутый’ словарь, то есть из {название: идентификатор} получится {идентификатор: название}. Эти функции неплохо сократили код.

 Как я уже сказал, использовалась библиотека requests, для того чтобы температуру и погоду можно было выбрать по городам, также на картинке можно видеть и все остальные критерии, по которым можно искать одежду.

Это самые главные функции, классы в моем коде. Но были и вещи, которые не были доделаны, например, предполагалось, что будет возможность искать одежду по ее цвету, также можно было бы попытаться сделать дизайн получше, возможность менять характеристику одежду в RandomClothes и другое. Но из-за того, что времени было не особо много, а сидеть за проектом днями напролет не хотелось, не все было сделано. Впрочем, я доволен и этим результатом, особенно тем, что использовал библиотеку requests, так как раньше я ничего и похожего не использовал.