

Критерии и правила для домашних работ

В задачах требуется использовать архитектуру «Модель-Вид». Все вводимые данные должны быть проверены на правильность. При нарушении формата данных в исходных файлах возможен один из трёх сценариев:

- прекращение работы программы;
- начало работы программы с пустыми данными, если это возможно;
- пропуск некорректной строки текстового файла.

Для получения оценки 4 или 5 должен быть разработан привлекательный внешний вид приложения: фон, шрифт текста, единый стиль оформления виджетов и т. п.

Список для проверки и самопроверки:

- Код организован с помощью архитектуры «Модель-Вид»
- Каждый класс находится в своём файле для удобства навигации по коду
 - Исключение: маленькие вспомогательные классы
- Для реализации всех функций программы используются модули Qt
 - Исключение: для открытия txt-файлов можно использовать функцию `open`, для открытия csv-файлов — модуль `pandas` или `csv`
- Для каждой функции в программе подобран подходящий механизм Qt
 - Для обработки событий используются сигналы и слоты
 - Если нужно отобразить список или таблицу с данными, то используются `QAbstractListModel + QListView` или `QAbstractTableModel + QTableView` (или `QTreeView`)
 - Верхнее меню организуется как `QMenuBar` или `QToolBar`
 - Работа с мышью и/или перетаскивание реализуется с помощью стандартных событий мыши и/или механизма `Drag&Drop`
 - Диаграммы и графики отображаются с помощью `Qt Charts` или `PyQtGraph`
- Для ввода данных используются подходящие виджеты, а не только простое поле ввода
 - Если нужно ввести число, то нужно воспользоваться `QSpinBox` (`QDoubleSpinBox`), `QSlider` или `QDial`
 - Если нужно ввести дату и/или время, нужно воспользоваться `QDateTimeEdit`, `QDateEdit` или `QTimeEdit`
 - Если нужно выбрать одно или несколько значений из фиксированного списка, то нужно воспользоваться `QComboBox`, `QCheckBox` или радиокнопками

- Если пользователь должен указать путь к файлу, то для выбора файла должен использоваться стандартный диалог с ограничениями по расширению выбираемого файла
- Для вывода данных не используются виджеты для ввода
- Есть обработка ошибок
 - Обрабатывается ситуация, когда в файле некорректные данные
 - Обрабатывается ситуация, когда пользователь ввёл пустые или некорректные данные
 - Если пользователь совершил ошибку, ему в графическом интерфейсе выводится информативное сообщение, как сделать правильно. Сообщение должно выделяться цветом или другими параметрами шрифта
- Главное окно приложения функционально: пользователь сразу может взаимодействовать с данными, не тратя действия на загрузку первоначальных данных или открытие другой страницы
 - Исключение: в некоторых приложениях пользователю сначала может выводиться страница для ввода логина и пароля
- Если приложение должно сортировать и/или фильтровать данные в таблице, то результат для удобства пользователя должен выводиться в той же таблице. Для этого можно воспользоваться механизмом `QSortFilterProxyModel`
- Открытие нового окна приложения или диалога можно заменить на открытие новой вкладки в главном окне. Произвольную навигацию по вкладкам для пользователя в большинстве случаев следует убрать, так как она мешает корректной работе приложения
- В приложениях-играх можно изменить предлагаемые действия пользователя на более интерактивные и требующие меньше нажатий (реакция на кнопки-стрелки на клавиатуре, реакция на колёсико и кнопки мыши и т. п.)
- Вы можете объяснить весь код приложения как на уровне архитектуры и взаимодействия классов, так и на уровне отдельных деталей