**Документация к компоненту course\_card.vue**

**Назначение**: Компонент course\_card.vue представляет собой интерактивную карточку курса с функционалом журнала успеваемости. Он включает таблицу с результатами студентов, возможность редактирования данных, фильтрацию, экспорт в Excel и управление файлами.

**Расположение**: src\components\course\_card.

**Структура компонента:**

**Template**

**Основная структура**:

* Внешний контейнер карточки с эффектами при наведении.
* Заголовок карточки с возможностью редактирования.
* Основное содержимое с фильтрами, таблицей и управляющими кнопками.

**Ключевые элементы**:

**Заголовок и тема**:

* Редактируемые поля для названия курса и темы.
* Кнопки редактирования и скрытия.

**Фильтры и поиск**:

* Выпадающие списки для выбора направления и группы.
* Поле поиска по ФИО.
* Выбор семестра.
* Кнопка экспорта в Excel.

**Таблица результатов**:

* Заголовки столбцов с фиксированными колонками (ФИО, Группа).
* Строка с максимальными баллами (редактируемая).
* Строки с результатами студентов (редактируемые).
* Подсчет общей суммы баллов для каждого студента.

**Управление файлами**:

* Загрузка и отображение прикрепленных файлов.
* Превью изображений.
* Удаление файлов.

**Модальные окна**:

* Добавление нового столбца.
* Добавление нового студента.

**Кнопки управления**:

* Добавить столбец.
* Добавить ученика.
* Сохранить изменения.
* Отмена редактирования.

**Script setup**

**Импорт:**

* import { ref, defineProps, defineEmits, computed, watch, inject, onMounted, nextTick } from "vue";
* import { cloneDeep } from "lodash";
* import \* as XLSX from "xlsx";

**Состояние компонента:**

* isCollapsed: Управляет состоянием свернутости/развернутости компонента. const isCollapsed = ref(false);
* isEditing: Управляет режимом редактирования. const isEditing = ref(false);
* errorMessage: Отображает сообщения об ошибках. const errorMessage = ref("");
* uploadedFiles: Хранит список загруженных файлов. const uploadedFiles = ref([]);
* isEditingGlobal: Глобальный режим редактирования. const isEditingGlobal = inject("isEditingGlobal");

**Props:**

* title: Заголовок журнала.
* topic: Тема журнала.
* container: Объект контейнера, для хранения данных.
* startEditing: Функция для начала редактирования.
* confirmDeleteContainer: Функция для подтверждения удаления контейнера.

**Refs:**

* titleText: Реактивная переменная для хранения заголовка.
* topicText: Реактивная переменная для хранения темы.
* columnOrder: Массив, определяющий порядок столбцов.
* resultsData: Массив данных о результатах студентов.
* maxScores: Объект, содержащий максимальные баллы для каждого столбца.
* maxScoresDraft: Черновик maxScores для редактирования.
* resultDraft: Черновик данных о результатах для редактирования.
* editingCell: Информация о редактируемой ячейке.
* searchQuery: Строка поискового запроса.
* filteredResults: Отфильтрованные данные о результатах.
* fioColumnWidth: Ширина столбца ФИО.
* groupColumnWidth: Ширина столбца группы.
* titleInput: Ссылка на элемент ввода заголовка.

**Методы:**

* addExamColumn: Добавляет столбец с экзаменационной оценкой, основываясь на диапазонах баллов. Рассчитывает оценку для каждого студента на основе его балла и переданных диапазонов оценок.
* calculateTotalScore: Вычисляет общую сумму баллов для студента.
* filterResults: Фильтрует результаты на основе поискового запроса.
* editCell: Начинает редактирование ячейки.
* editCellMaxScores: Начинает редактирование максимальных баллов.
* getGroupColumnLeft, getSumColumnLeft: Вычисляют позицию столбцов группы и суммы баллов.
* adjustInputWidth: Подстраивает ширину поля ввода заголовка.
* toggleCollapse: Переключает состояние свернутости/развернутости компонента.
* startEditing: Включает режим редактирования.
* saveChanges: Сохраняет изменения.
* cancelEditing: Отменяет изменения.
* exportToExcel: Экспортирует данные в Excel.

**Обработка событий жизненного цикла:**

* onMounted: Выполняется после монтирования компонента. Здесь определяется ширина столбцов ФИО и группы, а также создаются черновики данных.

**Watchers:**

* watch(props): Обновляет titleText и topicText при изменении пропсов.
* watch(titleText): Подстраивает ширину поля ввода заголовка при изменении titleText.

**Computed Properties:**

* totalMaxScore: вычисляет максимальную сумму баллов.

**Документация к компоненту literature.vue**

**Назначение:**

Компонент literature.vue представляет собой интерактивный блок, предназначенный для управления и отображения информации, связанной с успеваемостью студентов. Он предоставляет возможности указания границ оценок, выбора колонок для суммирования баллов, загрузки файлов и редактирования основных параметров.

**Расположение:** src/components/literature.vue

**Структура компонента:**

**Template**

**Основная структура:**

* Шаблон состоит из контейнера с гибкой структурой, содержащего заголовок, блок настроек оценок, выбор колонок для суммирования, загрузку файлов и кнопки сохранения и отмены.

**Ключевые элементы:**

**Заголовок:**

* Отображает название темы (topicText). Реализовано редактирование заголовка в режиме редактирования (isEditing). Используется изображение Rectangle.png для фона заголовка.

**Тема:**

* Отображает текст темы (topicText). Реализовано редактирование темы в режиме редактирования (isEditing).
* Блок настроек оценок: Позволяет задавать границы баллов для оценок “Отлично”, “Хорошо”, “Удовл.”, “Неуд.” (grades). Редактирование доступно только в режиме редактирования (isEditing).
* Выбор колонок для суммирования: Позволяет выбрать колонки, которые будут учитываться при расчете суммы баллов (localSelectedColumns). Редактирование доступно только в режиме редактирования (isEditing).

**Загрузка файлов:**

* Позволяет прикреплять файлы. Отображает превью изображений и ссылки на скачивание для других типов файлов. Добавление файла доступно только в режиме редактирования (isEditing).
* Кнопки управления (Сохранить, Отмена): Кнопки для сохранения изменений и отмены режима редактирования. Отображаются только в режиме редактирования (isEditing).
* Иконки редактирования и скрытия: Отображаются в правом верхнем углу, если включен глобальный режим редактирования (isEditingGlobal) и компонент не находится в режиме редактирования (!isEditing).

**Script setup**

**Импорт:**

* import { ref, defineProps, defineEmits, computed, watch, inject, onMounted } from "vue";
* import { useStorage } from '@vueuse/core';

**Состояние компонента:**

* isCollapsed: ref(false) - Управляет состоянием свернутости/развернутости компонента.
* isEditing: ref(false) - Управляет режимом редактирования компонента.
* errorMessage: ref("") - Хранит сообщение об ошибке.
* uploadedFiles: ref([]) - Хранит массив загруженных файлов (объекты с полями file и url).
* isEditingGlobal: inject("isEditingGlobal") - Получает глобальное состояние редактирования из родительского компонента (через inject).
* localSelectedColumns: ref([...props.selectedColumns]) - Локальная копия массива выбранных колонок.
* topicTitle: ref(props.topic) - Локальная копия заголовка темы.
* topicText: ref(props.topic) - Локальная копия текста темы.
* grades: ref(JSON.parse(JSON.stringify(initialGrades))) - Локальная копия границ оценок.

**Props:**

* topic: String (default: "ЖУРНАЛ") - Заголовок темы.
* totalMaxScore: Number (required) - Максимальная сумма баллов.
* columns: Array (default: []) - Массив доступных колонок для выбора.
* selectedColumns: Array (default: []) - Массив выбранных колонок (для суммирования баллов).
* grades: Array (default: [...]) - Массив объектов с границами оценок (Отлично, Хорошо, Удовл., Неуд.). Каждый объект содержит поля name, minScore, maxScore.
* maxScores: Object (default: {}) - Объект, содержащий максимальные баллы для каждой колонки. Ключ - название колонки, значение - максимальный балл.
* addExamColumn: Function (required) - Функция, вызываемая для добавления столбца с экзаменационной оценкой.
* students: Array (default: []) - Массив объектов с данными учеников.
* Emits:
* 'update:selectedColumns': Array - Генерируется при изменении выбранных колонок. Передает новый массив выбранных колонок.
* 'update:grades': Array - Генерируется при изменении границ оценок. Передает новый массив объектов с границами оценок.
* 'update:student-scores': Array - Генерируется при изменении оценок студентов.
* 'request-add-exam-column': - Генерируется, когда запрашивается добавление столбца экзамена.

**Computed Properties:**

* getLevel2Options: Array - Возвращает массив опций для второго выпадающего списка (поднаправления) в зависимости от выбранного направления.
* availableColumns: Array - Возвращает массив доступных колонок из пропса columns.
* selectedColumnsTotal: Number - Вычисляет сумму максимальных баллов для выбранных колонок.
* studentScores: Array - Вычисляет сумму баллов для каждого студента по выбранным колонкам.

**Методы:**

* toggleCollapse(): void - Переключает состояние свернутости/развернутости компонента.
* startEditing(): void - Включает режим редактирования.
* cancelEditing(): void - Отменяет изменения и выходит из режима редактирования.
* handleFileUpload(event: Event): void - Обрабатывает загрузку файлов.
* removeFile(index: number): void - Удаляет файл из списка загруженных файлов.
* setDefaultGradeValues(): void - Устанавливает значения по умолчанию для границ оценок при инициализации.
* loadGradesFromLocalStorage(): Array - Загружает оценки из localStorage.
* saveGradesToLocalStorage(grades: Array): void - Сохраняет оценки в localStorage.
* selectAllColumns(): void - Выбирает все доступные колонки.
* saveGradeBounds(): void - Выполняет валидацию границ оценок и сохраняет их.
* saveChanges(): void - Сохраняет изменения (включая валидацию).
* setGradesAndAddExamColumn(): void - Устанавливает границы оценок и отправляет запрос на добавление колонки экзамена.

**Обработка событий жизненного цикла:**

* onMounted(): void - Выполняется после монтирования компонента. Вызывает setDefaultGradeValues() для установки начальных значений границ оценок.

**Watchers:**

* watch(localSelectedColumns, (newVal) => ...) - Синхронизирует локальный массив выбранных колонок (localSelectedColumns) с пропсом selectedColumns через механизм emit.
* watch(grades, (newVal) => ...) - Синхронизирует локальный массив границ оценок (grades) с пропсом grades через механизм emit.
* watch(() => props.selectedColumns, (newVal) => ...) -Реагирует на изменения в массиве выбранных колонок, полученном из родительского компонента, и обновляет локальный массив localSelectedColumns, если они не совпадают.
* watch(studentScores, (newScores) => ...) - Отслеживает изменения в studentScores и генерирует событие 'update:student-scores' при их возникновении.
* watch(grades, (newGrades) => ...) - Отслеживает изменения в grades и генерирует событие 'update:grade-ranges' при их возникновении.
* watch(() => props.totalMaxScore, (newTotalMaxScore) => ...) - Отслеживает изменения в пропсе totalMaxScore и автоматически обновляет grades.value[0].maxScore (максимальное значение для оценки “Отлично”) при изменении totalMaxScore.

**Хранение данных (useStorage):**

* Использует @vueuse/core useStorage для хранения значений границ оценок в localStorage. Это позволяет сохранить значения между сессиями. Переменные: a1, a2, b1, b2, c1, c2, d1, d2.