СОДЕРЖАНИЕ

[1. ВЫПОЛНЕНИЕ ПРОСТЕЙШИХ ИЗМЕРЕНИЙ С ПОМОЩЬЮ МУЛЬТИМЕТРА И ИНДИКАТОРНОЙ ОТВЕРТКИ. 3](#_Toc169883788)

[2. СБОРКА ПК И ТЕСТИРОВАНИЕ ОСНОВНЫХ КОМПОНЕНТОВ. 5](#_Toc169883789)

[3. МИГРАЦИЯ ДИСКОВ ФОРМАТА MBR В ФОРМАТ GPT С СОХРАНЕНИЕМ ДАННЫХ И ВОССТАНОВЛЕНИЕ ЗАГРУЗКИ ОС. 7](#_Toc169883790)

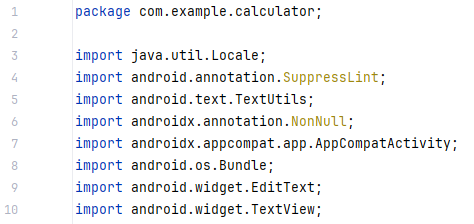
[4. МИГРАЦИЯ ОС НА НОВЫЙ ДИСК И РАЗБИЕНИЕ ДИСКОВ С ПОМОЩЬЮ ПРОГРАММЫ MINITOOL PARTITION WIZARD. 11](#_Toc169883791)

[5. ОСНОВЫ АДМИНИСТРИРОВАНИЯ WINDOWS SERVER. ЧАСТЬ 3. 14](#_Toc169883792)

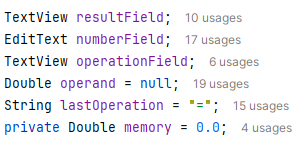
[ЗАКЛЮЧЕНИЕ 19](#_Toc169883793)

1. QWRFQWGE

Этот набор импортов является основой для создания пользовательского интерфейса в приложении Android. Он позволяет использовать различные компоненты и классы, необходимые для взаимодействия с пользователем и управления данными.



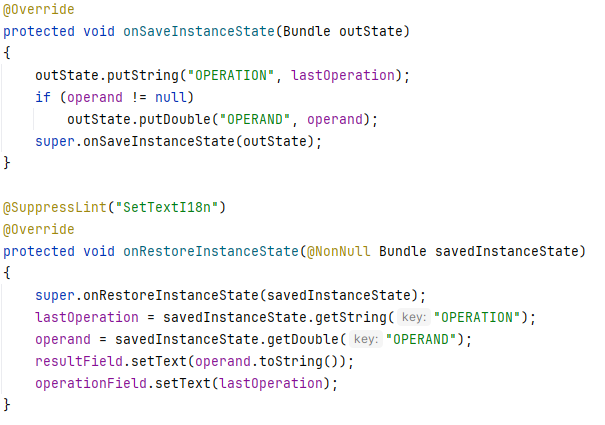
Данный класс представляет собой основу для калькулятора или подобного приложения, где пользователи могут вводить числа, выполнять операции и видеть результаты. В дальнейшем в классе, вероятно, будет добавлена логика для обработки ввода и выполнения математических операций. Это создаст функциональное приложение, которое будет взаимодействовать с пользователем и предоставлять ему необходимые вычисления.



В методе `onCreate` инициализируется интерфейс калькулятора и связываются элементы управления с соответствующими переменными. Устанавливаются обработчики кликов для кнопок операций, чисел и памяти, которые вызывают соответствующие методы для выполнения действий, таких как сложение, вычитание, вычисление корня и управление памятью. Это создает интерактивный интерфейс для пользователя. 😊



Методы `onSaveInstanceState` и `onRestoreInstanceState` сохраняют и восстанавливают состояние активности, сохраняя текущую операцию и операнд в `Bundle`.



Метод `onNumberClick` добавляет нажатую цифру или запятую в поле ввода, при этом предотвращая ввод нескольких запятых подряд, а также сбрасывает операнд, если предыдущая операция была равенством. Метод `onOperationClick` получает введённое число, проверяет, что оно не пустое, заменяет запятую на точку для корректного преобразования в число и вызывает `performOperation` с этим числом и выбранной операцией. В случае ошибки ввода очищает поле, а затем обновляет текущую операцию и отображает её.

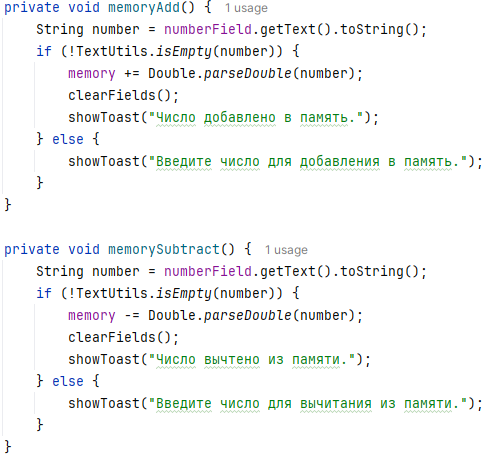


Метод `performOperation` выполняет арифметические операции над числом и обновляет результат. Если операнд ещё не установлен, он принимает значение текущего числа. Если последняя операция была равенством, она обновляется на новую. В зависимости от последней операции выполняется соответствующее вычисление. Если происходит деление на ноль, выводится ошибка и очищаются поля. Результат отображается с двумя знаками после запятой, а поле ввода очищается для следующего числа.



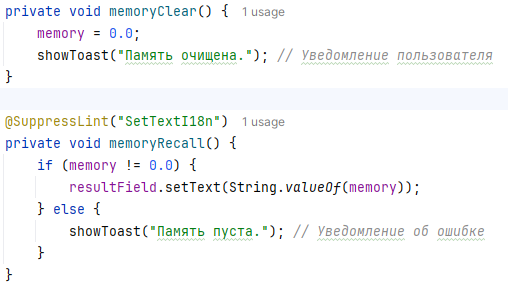
Методы `memoryAdd` и `memorySubtract` предназначены для работы с памятью калькулятора. В первом методе, `memoryAdd`, сначала считывается текущее значение из поля ввода. Если это значение не пустое, оно преобразуется в число и добавляется к текущему значению памяти. После этого вызывается метод `clearFields`, который очищает поле ввода, и пользователю отображается уведомление о том, что число успешно добавлено в память. Если же поле ввода пустое, выводится сообщение с просьбой ввести число для добавления.

Во втором методе, `memorySubtract`, процесс аналогичен. Сначала считывается значение из поля ввода. Если оно не пустое, то это значение вычитается из текущего значения памяти. После выполнения операции также очищается поле ввода, и пользователю показывается уведомление о том, что число вычтено из памяти. Если поле ввода пустое, выводится сообщение, призывающее пользователя ввести число для вычитания.

Таким образом, оба метода обеспечивают удобное и безопасное взаимодействие с памятью калькулятора, информируя пользователя о результатах операций и предотвращая ошибки, связанные с пустым вводом.

Метод `memoryRecall` отвечает за вывод значения, сохранённого в памяти калькулятора, на экран результата. Если значение памяти не равно нулю, оно преобразуется в строку и устанавливается в поле `resultField`, чтобы пользователь увидел сохранённое число. Если же память равна нулю, считается, что она пуста, и вместо вывода числа показывается уведомление с сообщением «Память пуста.» — это помогает информировать пользователя о состоянии памяти.

Метод `showToast` служит для отображения кратких всплывающих сообщений (тостов) на экране. Он принимает строку с текстом сообщения и показывает её пользователю на короткое время, используя стандартный механизм Android для уведомлений. Такой подход удобен для информирования пользователя о результатах операций или ошибках без необходимости использовать дополнительные диалоговые окна.



Метод `calculateSqrt` предназначен для вычисления квадратного корня из числа, введённого пользователем в поле `numberField`. Сначала метод проверяет, не является ли поле ввода пустым. Если оно пустое, выполнение метода просто прекращается, и никакие действия не выполняются.

Далее, если поле не пустое, происходит замена запятых на точки, чтобы обеспечить корректное преобразование строки в число. Затем метод пытается преобразовать строку в число типа `double`. Если преобразование не удаётся, выбрасывается исключение `NumberFormatException`, и поле ввода очищается.

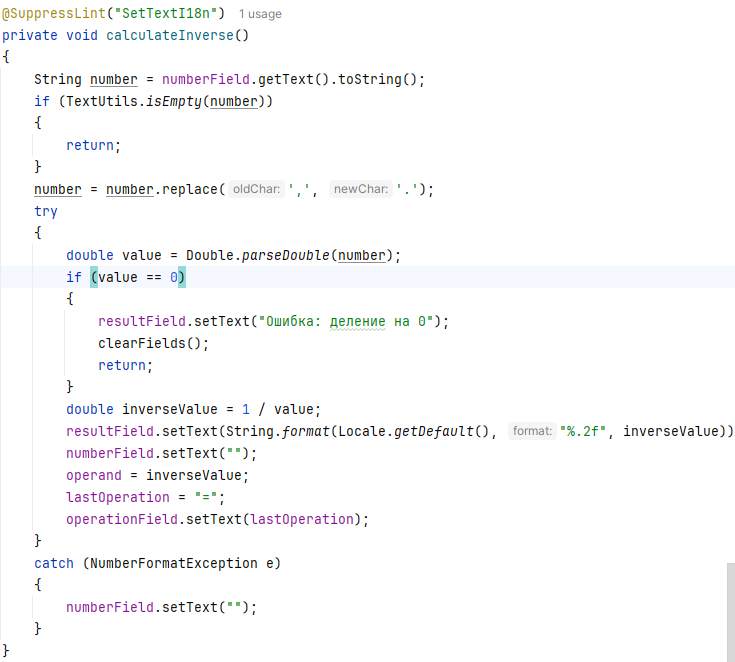
Если преобразование прошло успешно, метод проверяет, не является ли значение отрицательным. Если значение меньше нуля, в поле результата (`resultField`) выводится сообщение об ошибке, указывающее на то, что нельзя извлечь квадратный корень из отрицательного числа. После этого поле ввода очищается, и выполнение метода завершается.

Если значение положительное, вычисляется квадратный корень с помощью функции `Math.sqrt`. Результат округляется до двух десятичных знаков и отображается в поле результата. Поле ввода также очищается, а переменная `operand` обновляется значением квадратного корня. Переменная `lastOperation` устанавливается в «=», и это значение отображается в поле операций (`operationField`).

Таким образом, метод обеспечивает безопасное и корректное вычисление квадратного корня, информируя пользователя о возможных ошибках и выводя результат в удобочитаемом формате.



Метод `calculateInverse` вычисляет обратное значение числа, введённого пользователем в поле `numberField`. Сначала он проверяет, пусто ли поле ввода, и если да, то просто завершает выполнение. Если поле не пустое, метод заменяет запятые на точки, чтобы корректно преобразовать строку в число. Затем происходит попытка преобразования строки в число типа `double`. Если преобразование не удаётся, вызывается исключение `NumberFormatException`, и поле ввода очищается. Если число успешно получено, метод проверяет, не равно ли оно нулю. В случае нуля в поле результата появляется сообщение об ошибке, что деление на ноль невозможно, после чего поле ввода очищается и выполнение метода завершается. Если число отлично от нуля, вычисляется его обратное значение, которое округляется до двух десятичных знаков и отображается в поле результата. После этого поле ввода очищается, переменная `operand` обновляется новым значением, а переменная `lastOperation` устанавливается в знак равенства, который отображается в поле операций. Таким образом, метод обеспечивает корректное вычисление обратного значения с обработкой ошибок и информированием пользователя.



Метод `deleteLastCharacter` отвечает за удаление последнего символа из текста, введённого пользователем в поле `numberField`. Сначала метод получает текущий текст из поля. Если текст не пустой, он обновляет содержимое поля, удаляя последний символ с помощью метода `substring`. Это позволяет пользователю корректировать ввод, удаляя ненужные цифры или символы.

Метод `clearFields` предназначен для очистки всех полей ввода и вывода. Он устанавливает текст в поле `numberField` и `resultField` на пустую строку, а также сбрасывает переменные `operand` и `lastOperation`. Переменная `lastOperation` устанавливается в знак равенства, который затем отображается в поле операций (`operationField`). Это обеспечивает удобство работы с калькулятором, позволяя пользователю начинать новый ввод без остаточных данных.

