1. **СОХРАНЕНИЕ СОСТОЯНИЯ ПРИЛОЖЕНИЯ**

*Задание 1*

Сформируем layout-файл главного Activity (рис. 1.1) со следующим содержимым: в корневом контейнере ConstraintLayout разместим элемент EditText (идентификатор nameBox) для ввода произвольного текста (например, имени) и кнопку для этого ввода (saveBtn, надпись «Сохранить»); а также элемент TextView (nameView) и вторую кнопку (getBtn, надпись «Восстановить») для получения сохранённого текста.

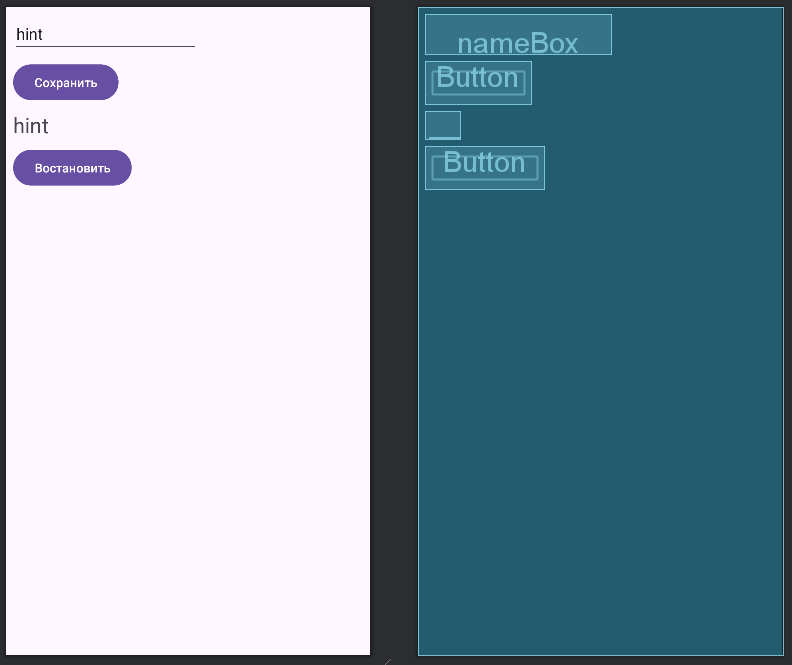


Рисунок 1.1 – Внешний вид разметки приложения

Реализуем в классе главного Activity методы сохранения и восстановления текста рисунок 1.2.

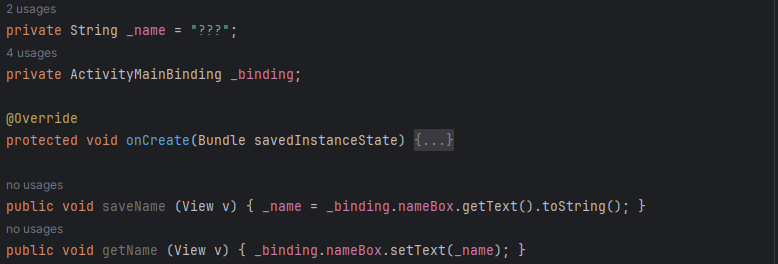


Рисунок 1.2 – Содержимое файла MainActivity.java

Запустим приложение, введем какой-либо текст, сохраним его и восстановим в TextView. Перейдя к альбомному режиму – TextView окажется пустым, несмотря на то что в нём было значение. По кнопке «Восстановить» попробуем снова получить ранее сохранённое значение – оно окажется утраченным.

Чтобы избежать подобных ситуаций следует сохранять и восстанавливать состояние Activity с помощью методов onSaveInstanceState и onRestoreInstanceState. Изменим код MainActivity рисунок 1.3.

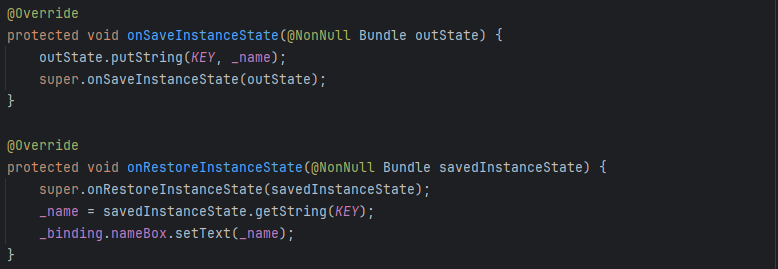


Рисунок 1.3 – Добавленные методы в MainActivity.java

Запустим приложение, проверим правильность работы при поворотах экрана. Приложение предоставлено на рисунке 1.4.

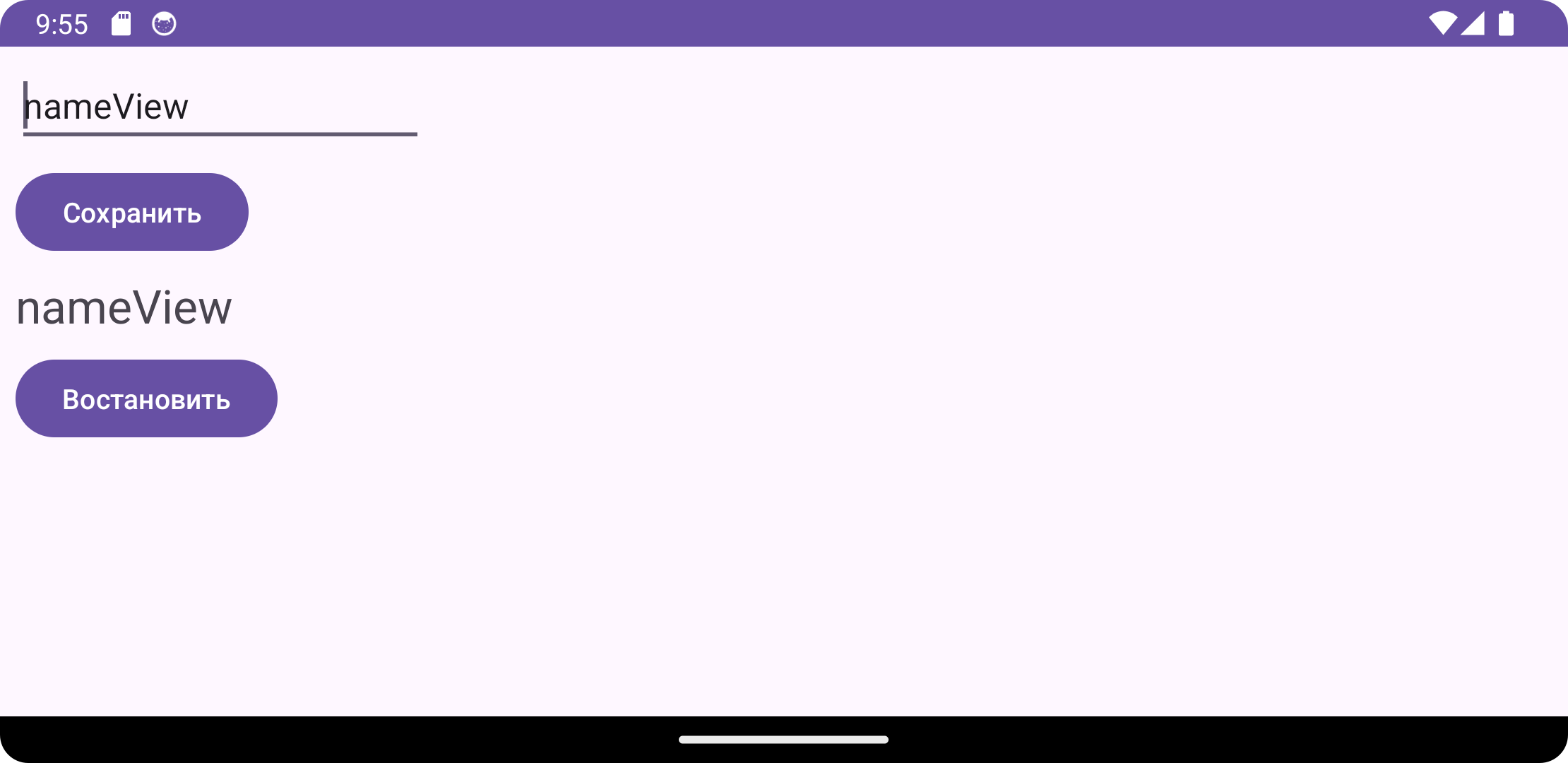
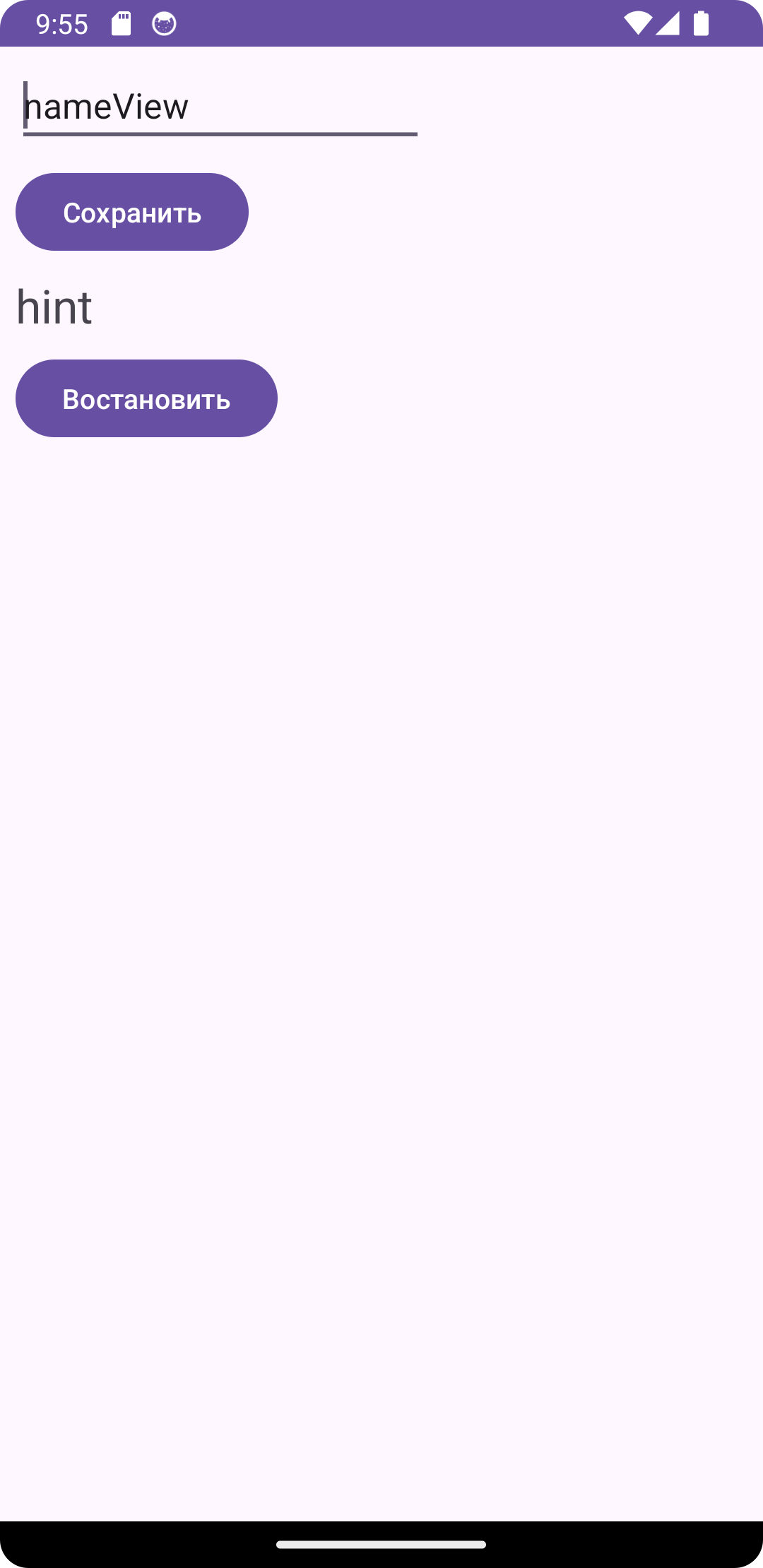


Рисунок 1.4 – Приложение запущенное на эмуляторе

*Задание 2*

Добавим в проект класс User и сформируем его код в рисунок 2.1.



Рисунок 2.1 – Содержимое файла User.java

Сформирем layout-файл главного Activity (рис. 2.2) со следующим содержимым: в корневом контейнере ConstraintLayout разместим два элемента EditText (идентификаторы nameBox и yearBox) для ввода имени и возраста человека (с соответствующими подсказками в полях ввода) и кнопку для этого ввода (saveBtn, надпись «Сохранить»); а также элемент TextView (dataView) и вторую кнопку (getBtn, надпись «Получить» или «Восстановить») для получения сохранённого текста.

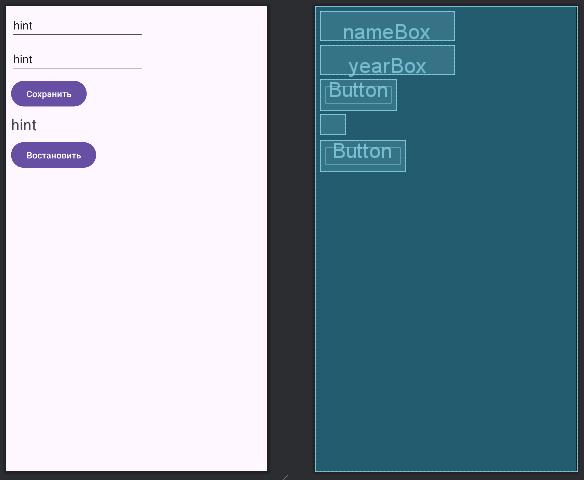


Рисунок 2.2 – Внешний вид разметки приложения

Реализуем в классе главного Activity методы сохранения и восстановления состояния Activity рисунок 2.3.

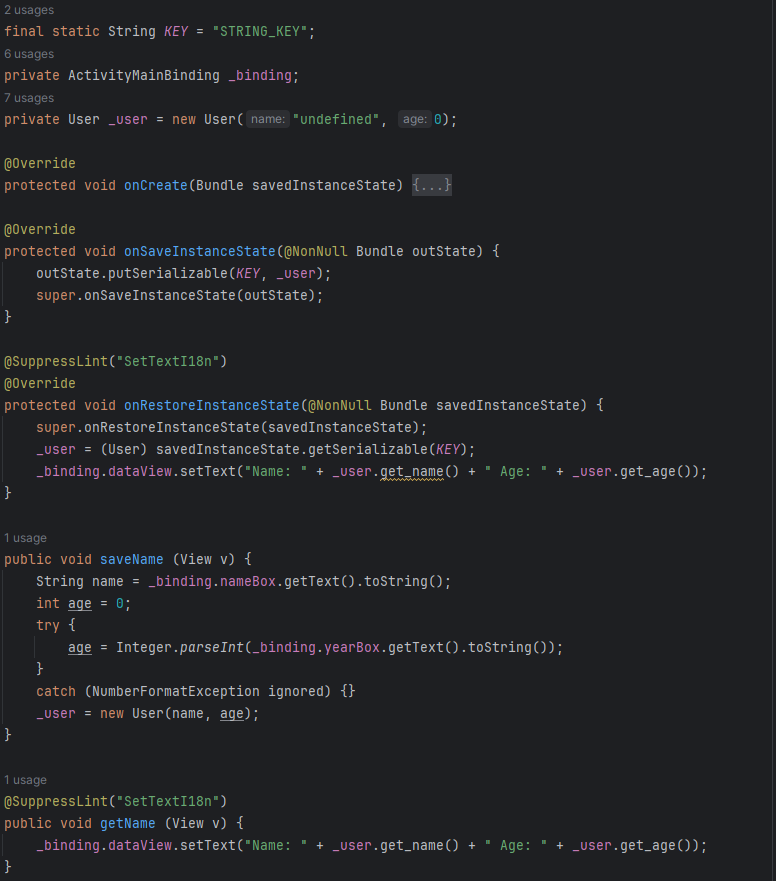


Рисунок 2.3 – Содержимое файла MainActivity.java

Запустим приложение, проверим правильность работы при поворотах экрана. Приложение предоставлено на рисунке 2.4.

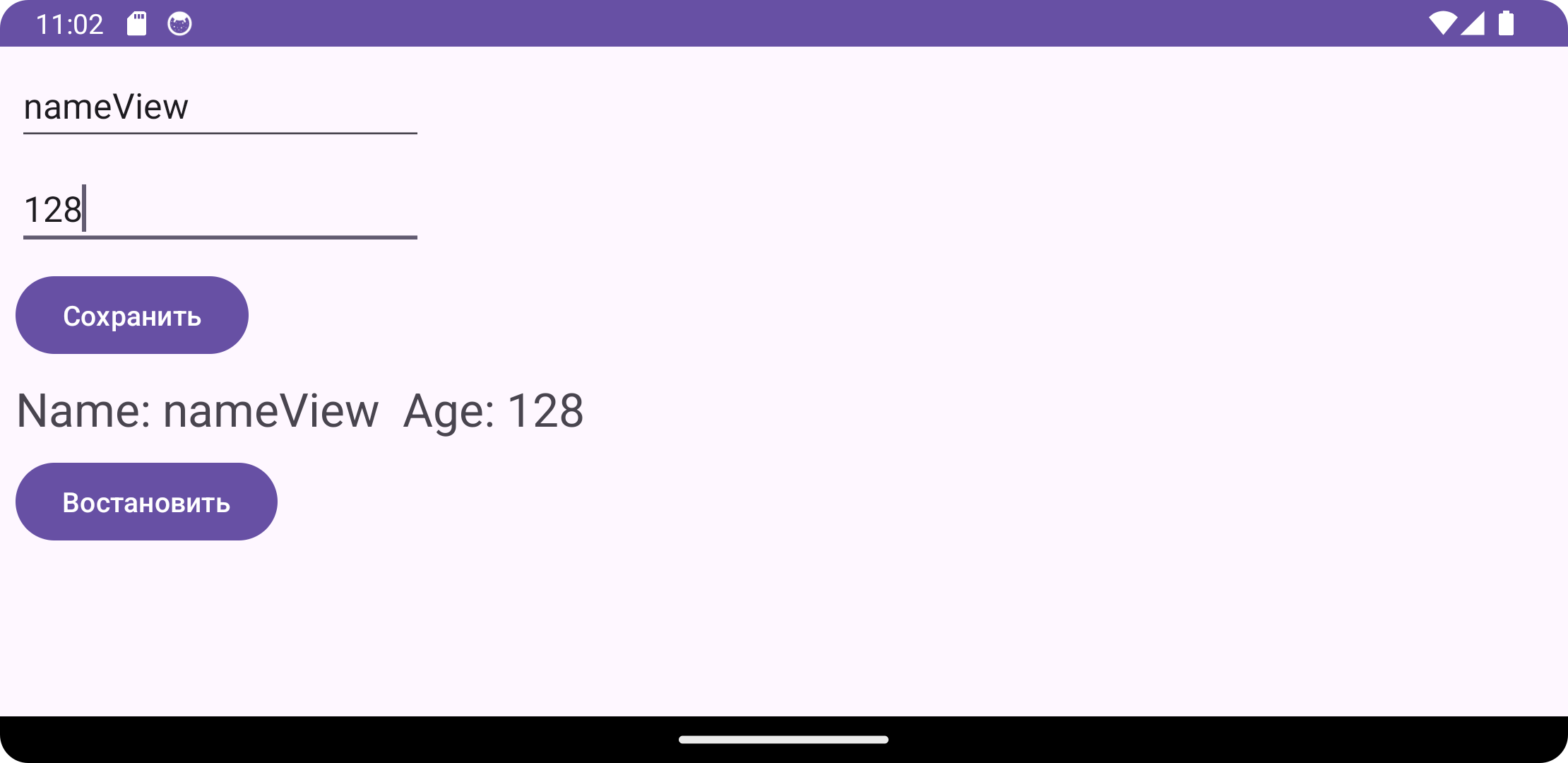
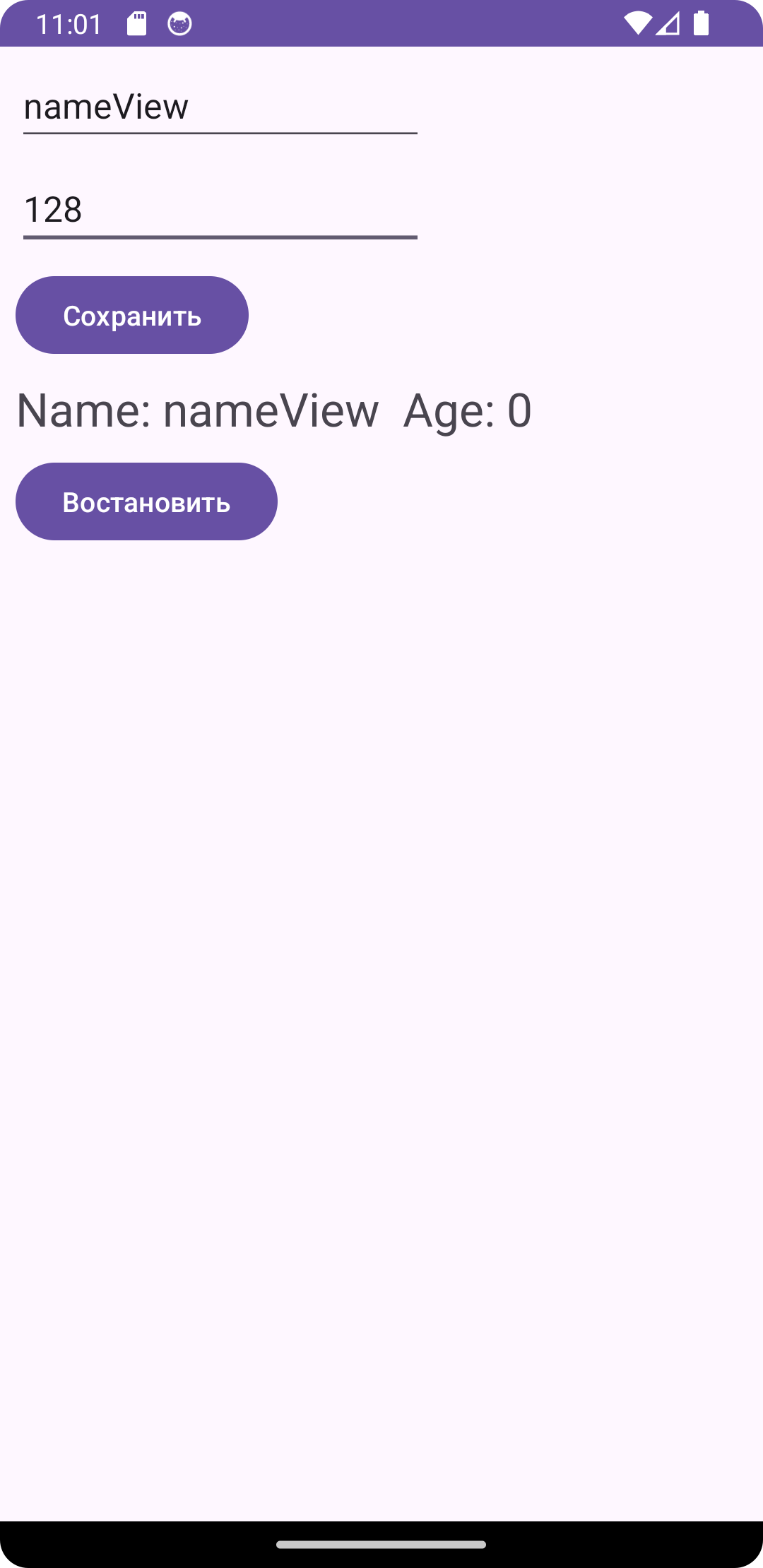


Рисунок 2.4 – Приложение запущенное на эмуляторе

1. **SHAREDPREFERENCES ДЛЯ СОЗДАНИЯ И ПОЛУЧЕНИЯ НАСТРОЕК**

*Задание 3*

Сформируем layout-файл главного Activity (рис. 3.1) со следующим содержимым: в корневом контейнере ConstraintLayout разместим элемент EditText (идентификатор nameBox) для ввода имени человека (с соответствующей подсказкой в поле ввода) и кнопку для этого ввода (saveButton, надпись «Сохранить»); а также элемент TextView (nameView) и вторую кнопку (getButton, надпись «Восстановить») для получения сохранённого имени.

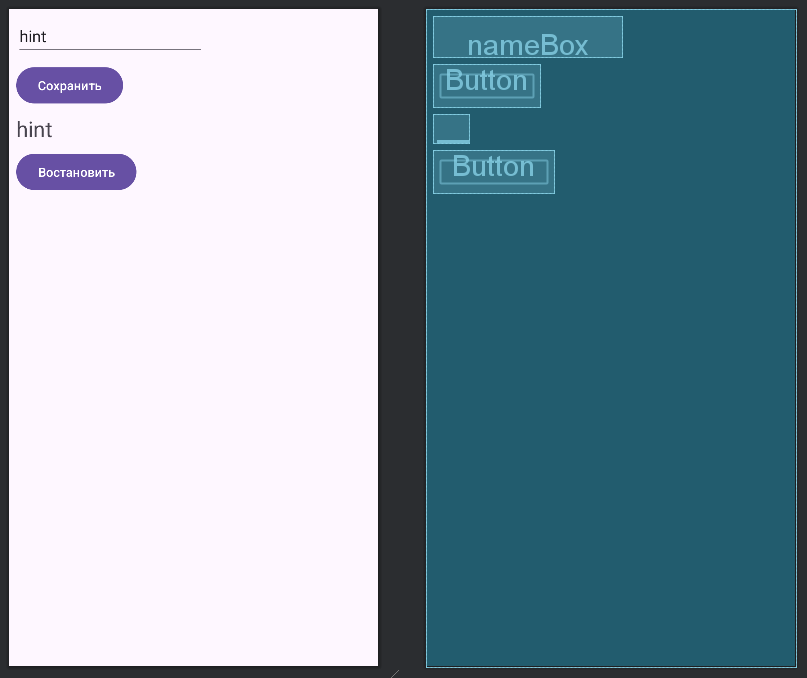


Рисунок 3.1 – Внешний вид разметки приложения

Определим методы обработки нажатия кнопок в классе MainActivity. А также переопределим методы onCreate и onPause, чтобы при выходе из приложения значение текстового поля автоматически сохранялось, а при запуске – восстанавливалось. Код предоставлен на рисунке 3.2.



Рисунок 3.2 – Содержимое файла MainActivity.java

Запустим приложение. Нажмите кнопку «Восстановить»: при отсутствии настроек при попытке их получить, приложение выведет значение по умолчанию. Запущенное приложение на эмуляторе предоставлено на рисунке 3.3.

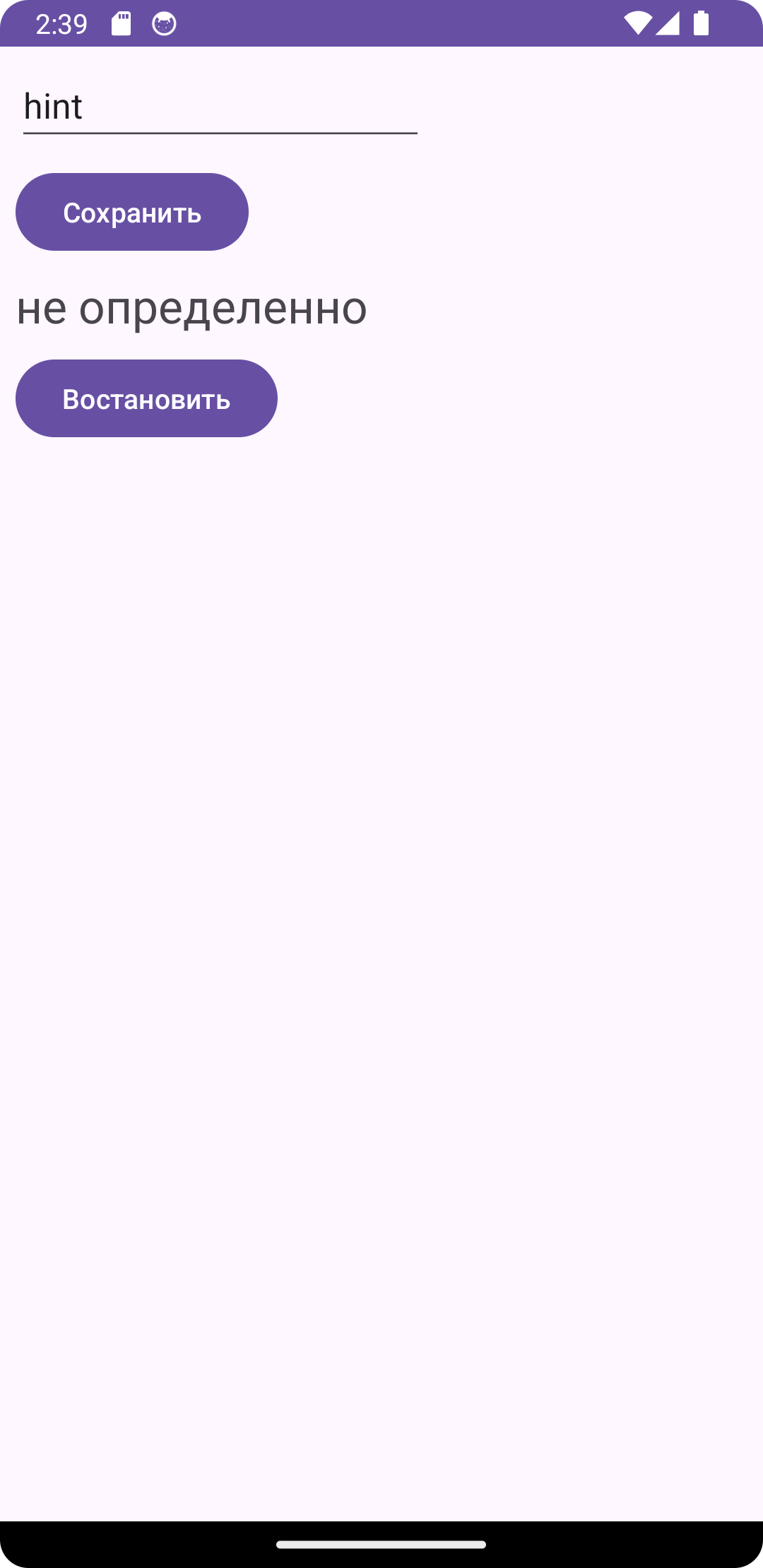


Рисунок 3.3 – Приложение запущенное на эмуляторе

1. **РАБОТА С ФАЙЛОВОЙ СИСТЕМОЙ**

**Чтение и запись файлов во внутреннем хранилище**

*Задание 4*

Сформируем layout-файл главного Activity (рис. 4.1) со следующим содержимым: в корневом контейнере ConstraintLayout разместим элемент EditText (идентификатор editor) для ввода текста (с соответствующей подсказкой в поле ввода) и кнопку для сохранения ввода в файл (save\_text, надпись «Записать»); а также элемент TextView (text) и вторую кнопку (open\_text, надпись «Прочитать») для получения из файла сохранённого текста.

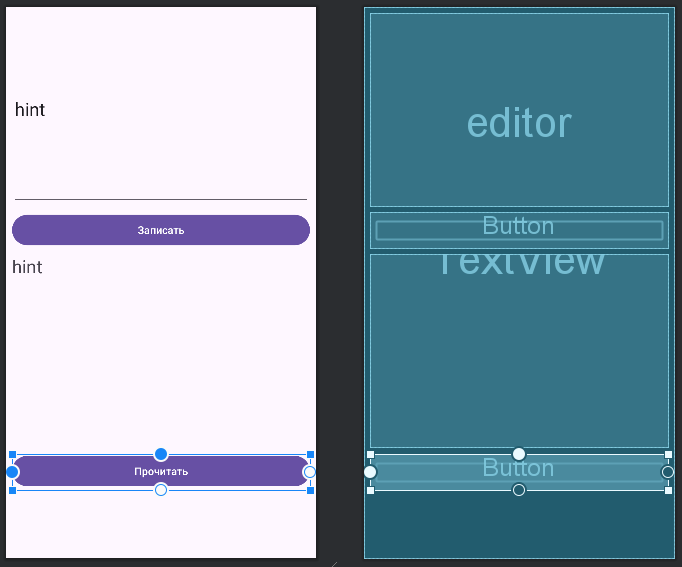


Рисунок 3.1 – Внешний вид разметки приложения

Определим в главном Activity обработчики кнопок в соответствии рисунок 3.2.

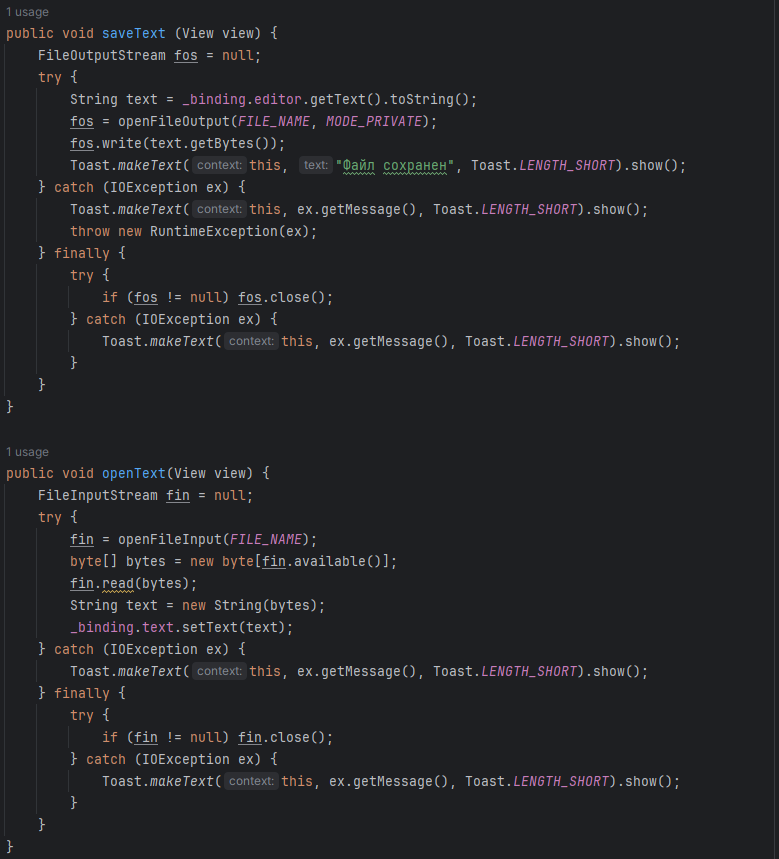


Рисунок 4.2 – Методы, отвечающие за отработку нажатий по кнопкам

Запущенное приложение на эмуляторе предоставлено на рисунке 4.3.

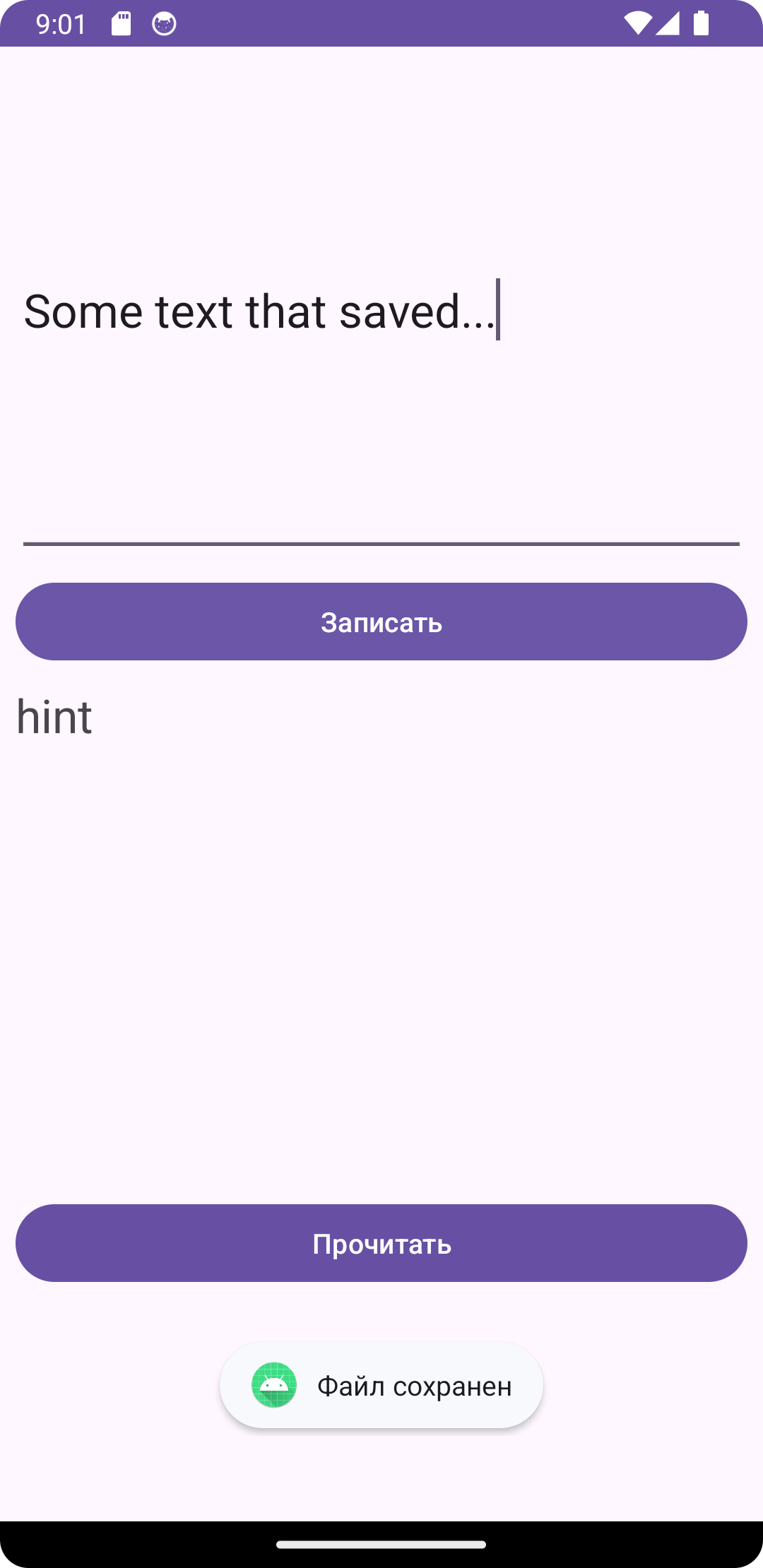
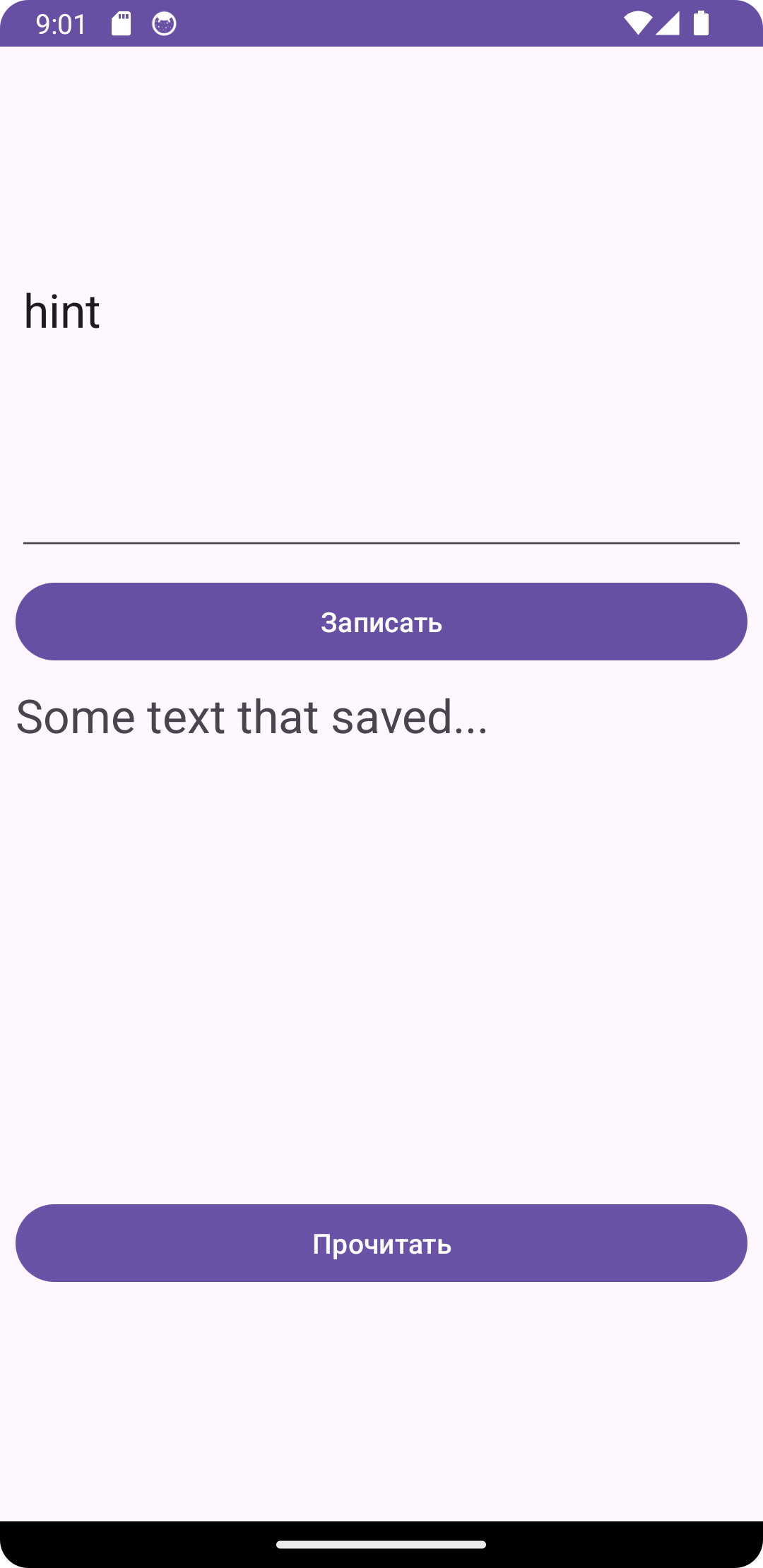
 

Рисунок 4.3 – Приложение запущенное на эмуляторе

**Работа с файлами на внешнем накопителе**

Выше был рассмотрен вариант работы с файлами из каталога приложения. По умолчанию такие файлы доступны только самому приложению. Возможно также помещать и работать с файлами из внешнего хранилища устройства. Это предоставит доступ к данным файлам и другим программам. В предыдущем проекте отредактируйте код главного Activity для работы с файлами на внешнем накопителе рисунок 4.4.

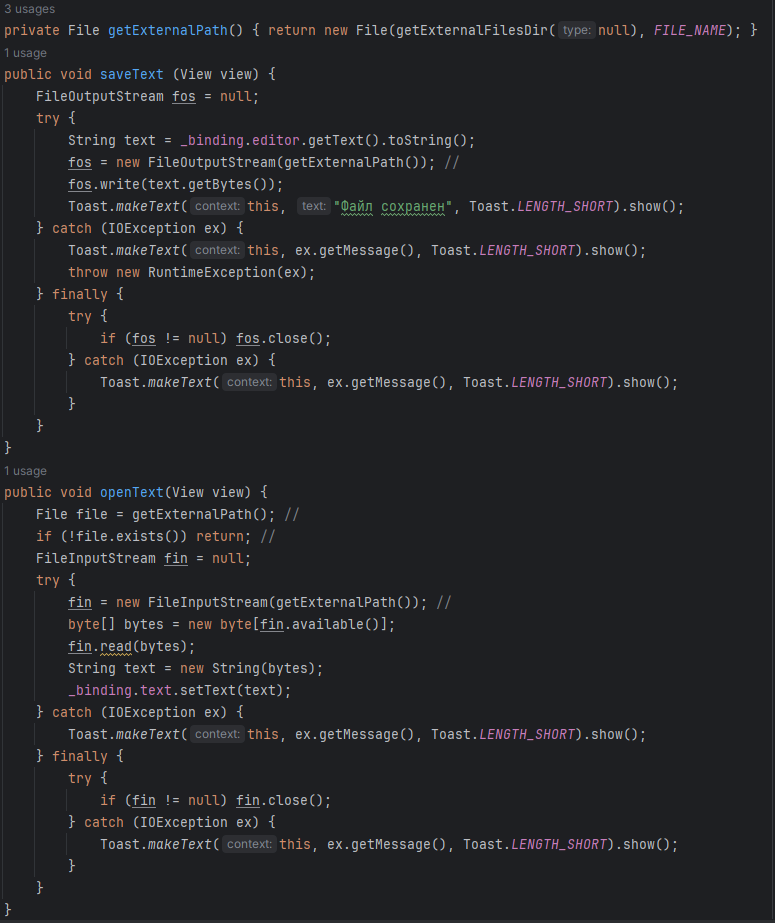


Рисунок 4.4 – Изменённые методы

После запуска нашего приложения и нажатия на кнопку “Сохранить” сохранённый файл мы сможем найти в папке приложения в .../data рисунок 4.5.

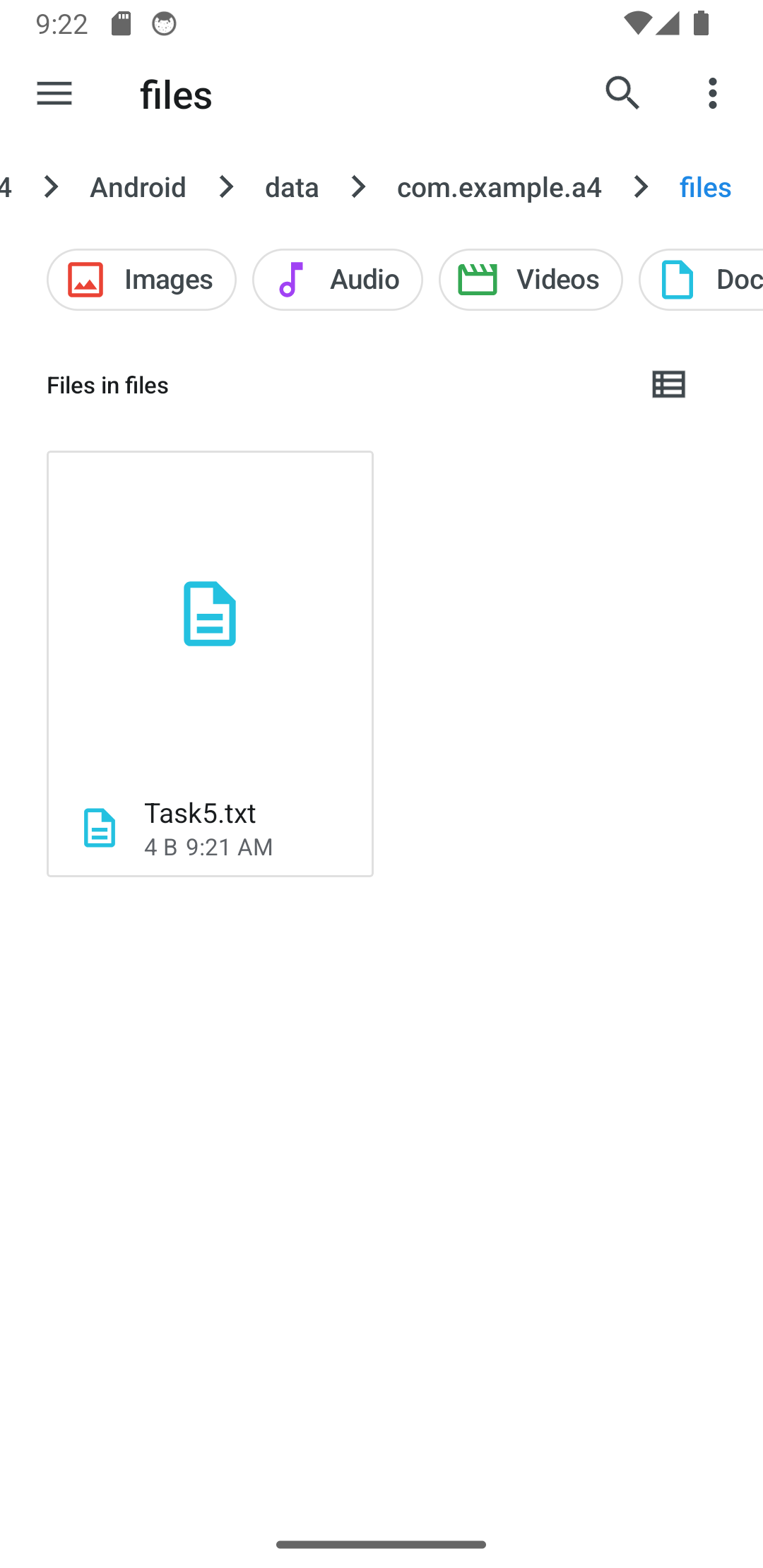


Рисунок 4.5 – Путь к нашему файлу в проводнике

1. **ФОРМАТ JSON**

*Задание 5*

Создадим новый класс Cat рисунок 5.1.

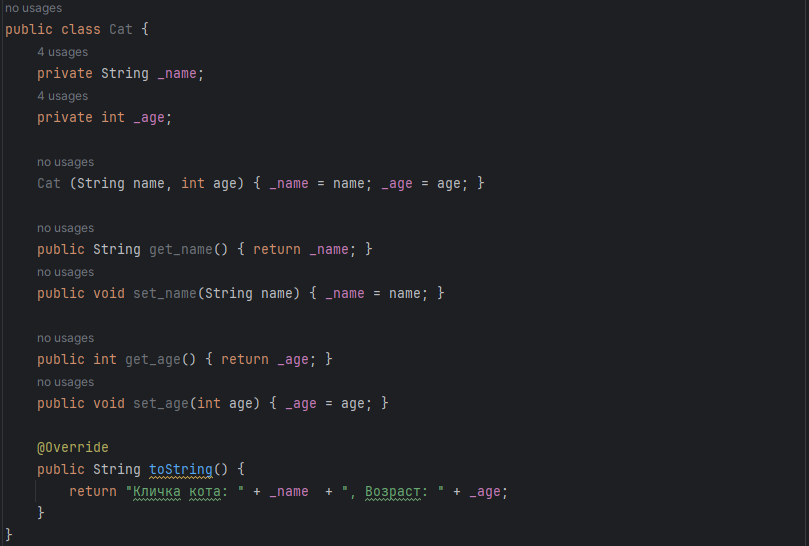


Рисунок 5.1 – Содержимое класса Cat

Объекты этого класса будем сериализовать в формат json и затем обратно десериализовать из файла в объекты.

Для работы с форматом json добавьте в проект класс myJSON рисунок 5.2.



Рисунок 5.2 – Файл myJSON.java

Сформируем layout-файл главного Activity (рис. 5.3) со следующим содержимым: в корневом контейнере ConstraintLayout разместите два элемента EditText (идентификаторыnameText и ageText) для ввода текста (с соответствующей подсказкой в поле ввода) и кнопку для добавления ввода в список (addButton, обработчик add); две кнопки (saveButton и openButton, обработчики save и open) для сериализации (записи в json-файл) и десериализации из файла соответственно; а также элемент ListView (идентификатор list) для вывода списка на экран.

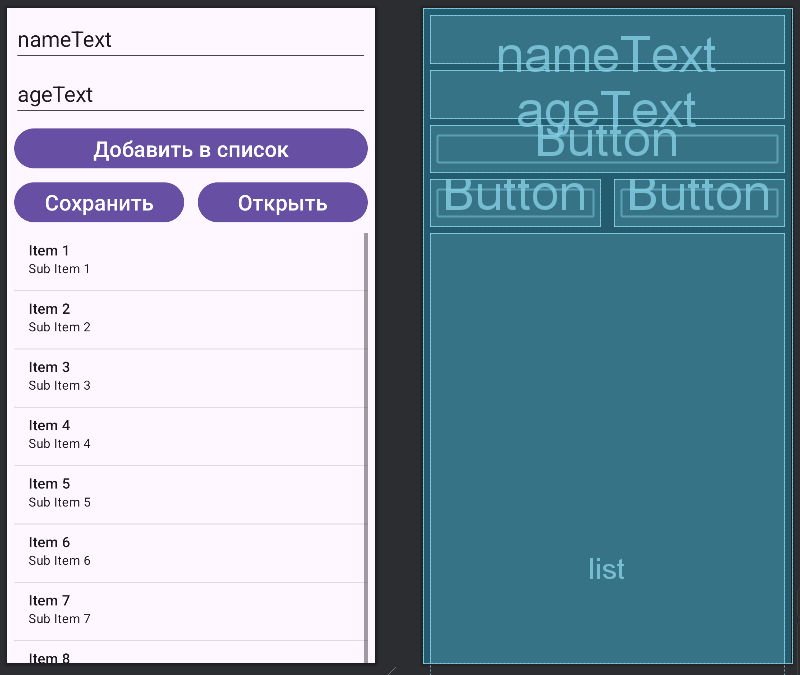


Рисунок 3.1 – Внешний вид разметки приложения

Определим код главного Activity рисунок 5.4.



Рисунок 5.3 – Файл Main.Activity.java

Запущенное приложение на эмуляторе предоставлено на рисунке 5.4.

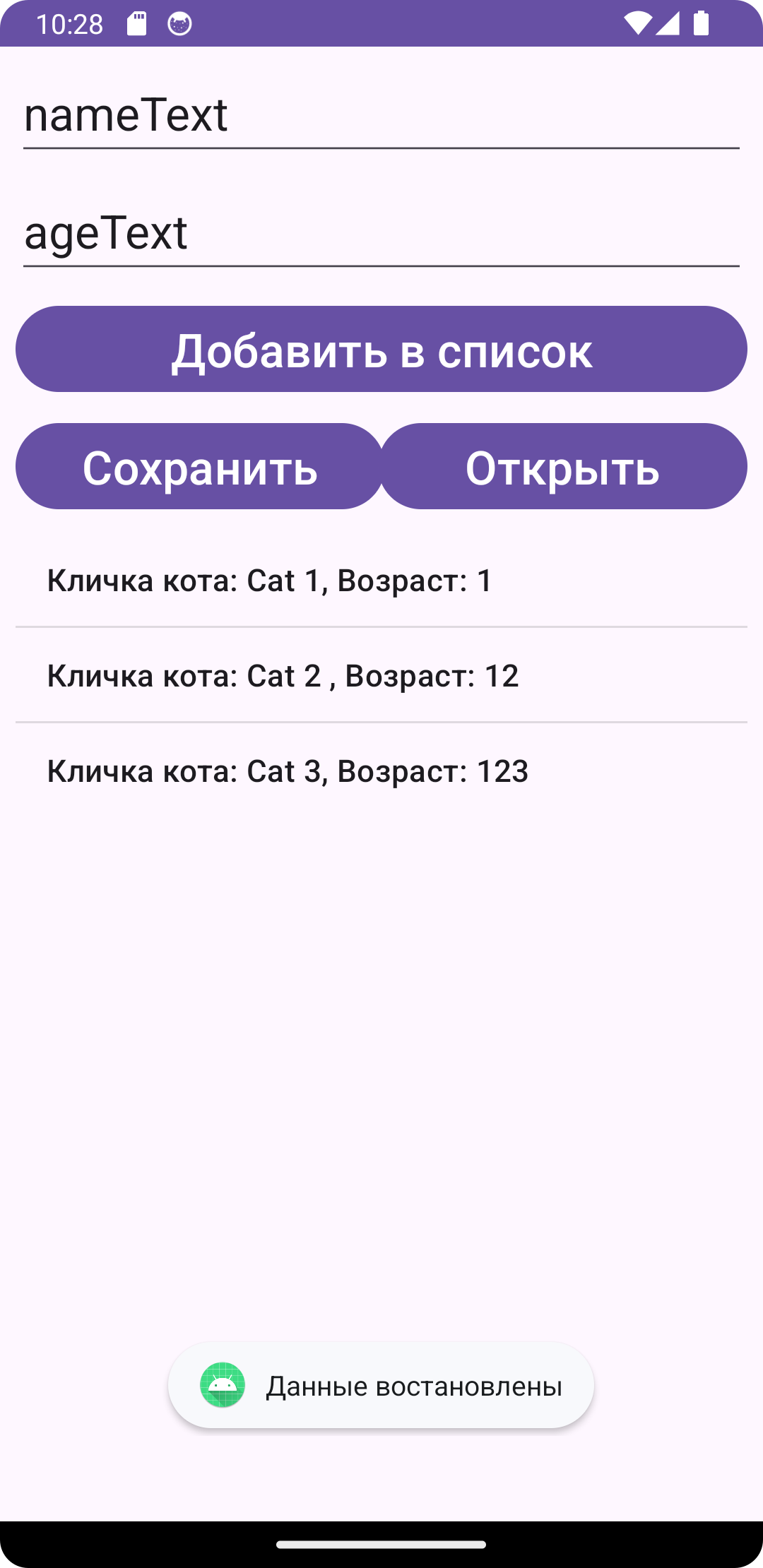
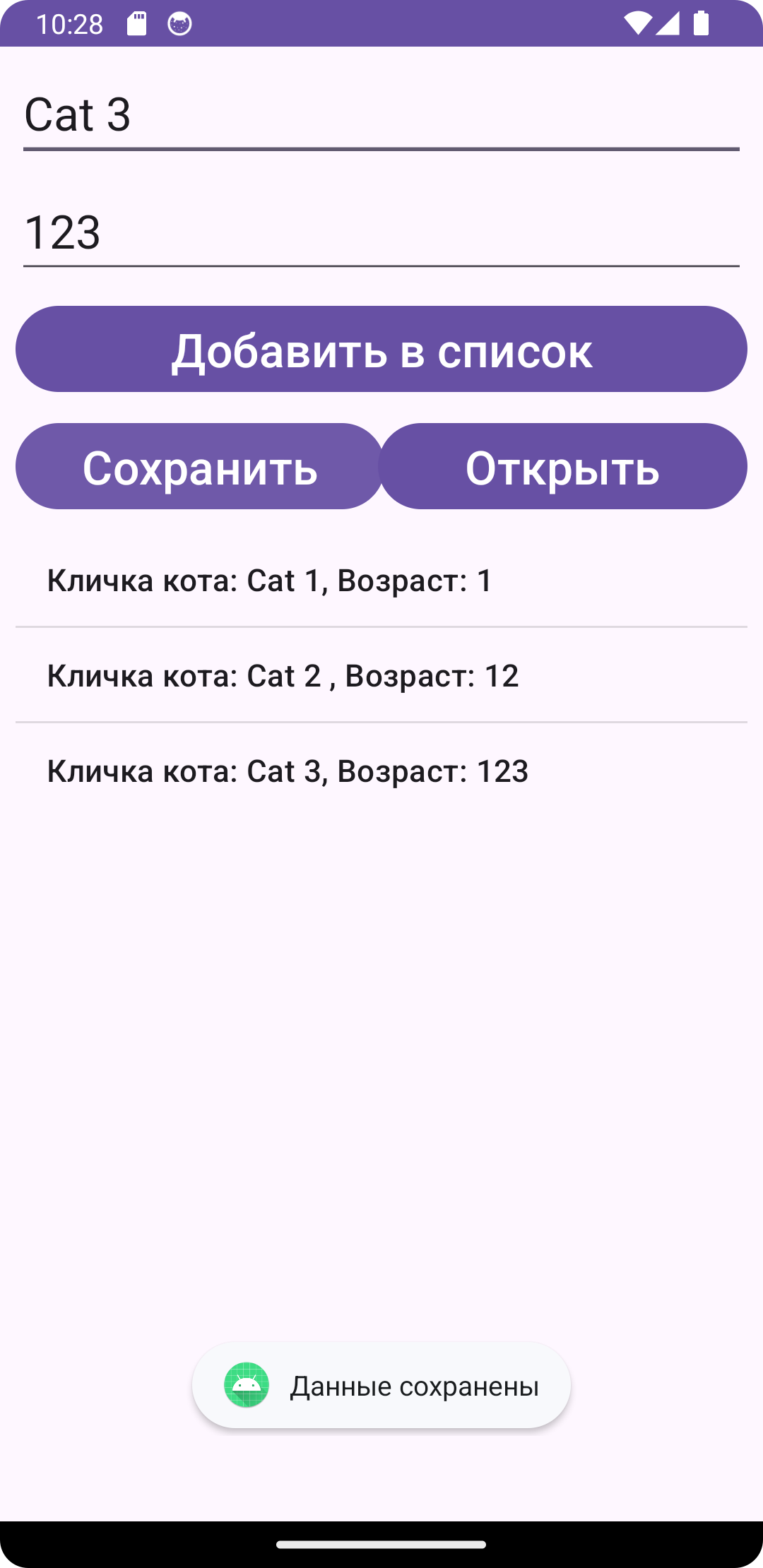


Рисунок 5.4 – Результат работы приложения

1. **ПРАКТИЧЕСКИЕ ЗАДАНИЯ**

*Задание 6*