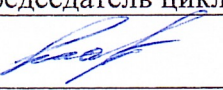



Утверждаю	Согласовано
Председатель цикловой комиссии	Заместитель директора по УР
 Е.Ю. Мартынова	 О.Ю. Корнеева
от 31.01.2024г.	от 31.01.2024г.

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРОВЕДЕНИЮ ЗАНЯТИЯ:
«Разработка мобильного приложения в среде 1С: Предприятие»**

Разработчик: преподаватель
ГБПОУ МО «Луховицкий авиационный техникум»
Колесников Никита Юрьевич

2024г.



Выполнение задания (60 мин)

1. *Подробная инструкция по выполнению задания.* Участник выполняет пробу в программе 1С: Предприятие 8.3.

Конфигурация представляет собой описание структуры данных, которые пользователь будет использовать в режиме работы 1С: Предприятие. Кроме этого, конфигурация описывает всевозможные алгоритмы обработки этих данных, содержит информацию о том, как эти данные должны будут выглядеть на экране и на принтере и т.д. В дальнейшем платформа «1С: Предприятие 8» на основании этого описания сама создаст базу данных, которая будет иметь необходимую структуру и предоставит пользователю возможность работать с этой базой данных.

Для начала работы нам необходимо открыть конфигурацию, с которой мы будем работать.

Для этого запустите ярлык Технологической платформы 1С Предприятие 8 расположенный на вашем рабочем столе.





Рисунок 1. Ярлык 1С Предприятие.

В открывшемся окне нажмите кнопку добавить, для создания новой информационной базы.

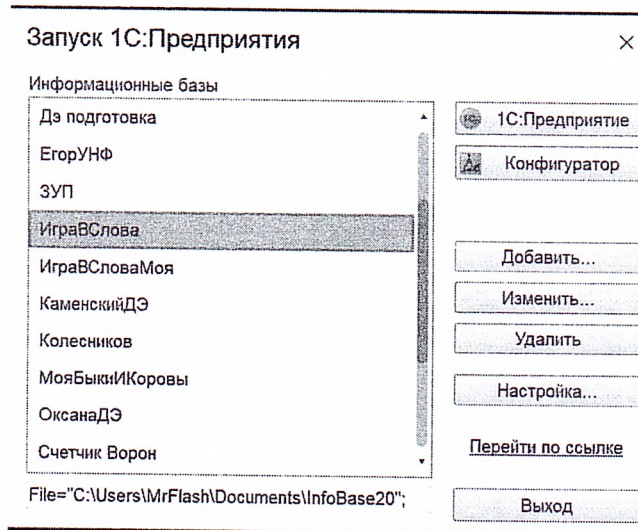


Рисунок 2. Окно выбора конфигурации.

Далее выберите пункт создание новой информационной базы.

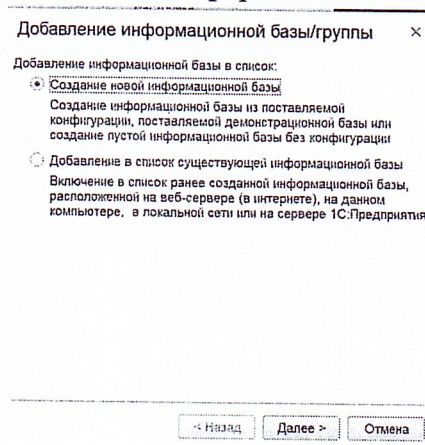


Рисунок 3. Создание новой информационной базы

Затем выбираем вариант «Создание информационной базы без конфигурации...» и нажимаем «Далее»:

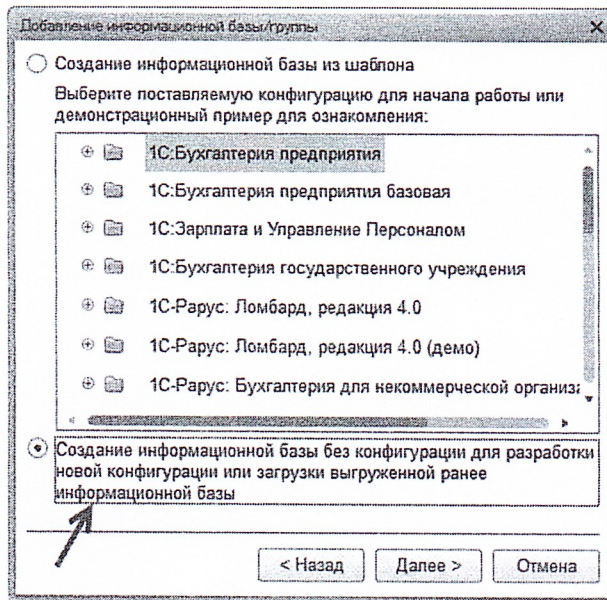


Рисунок 4. Создание информационной базы без конфигурации
Указываем название нашей базы:

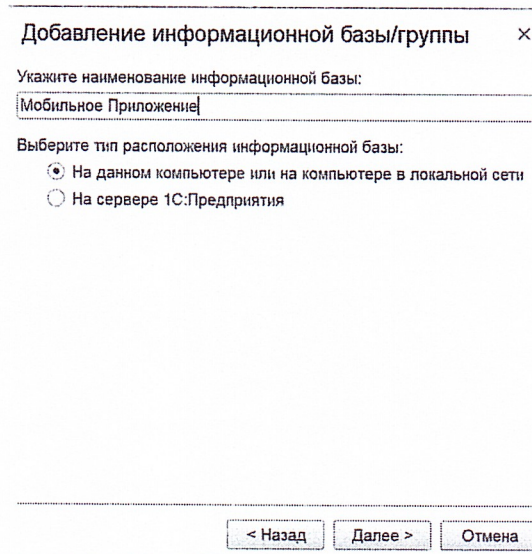


Рисунок 5. Название базы
Указываем каталог, в котором будет храниться наша база:

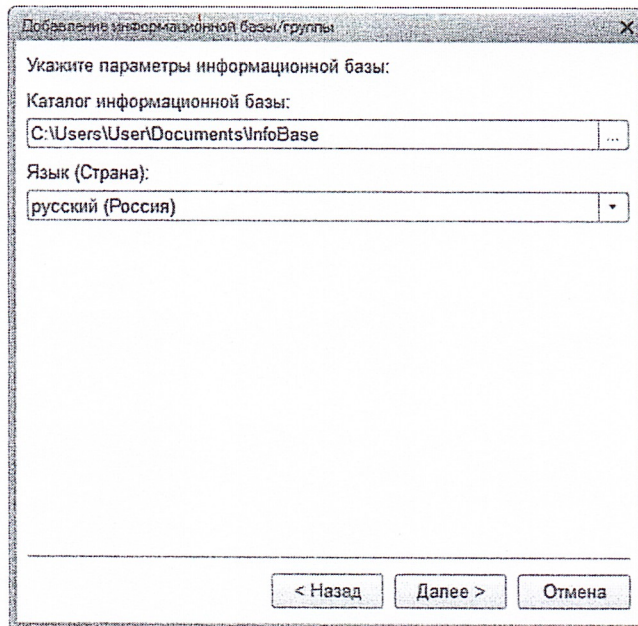


Рисунок 6. Указание каталога

Типы запуска оставляем по умолчанию, нажимаем «Готово»:

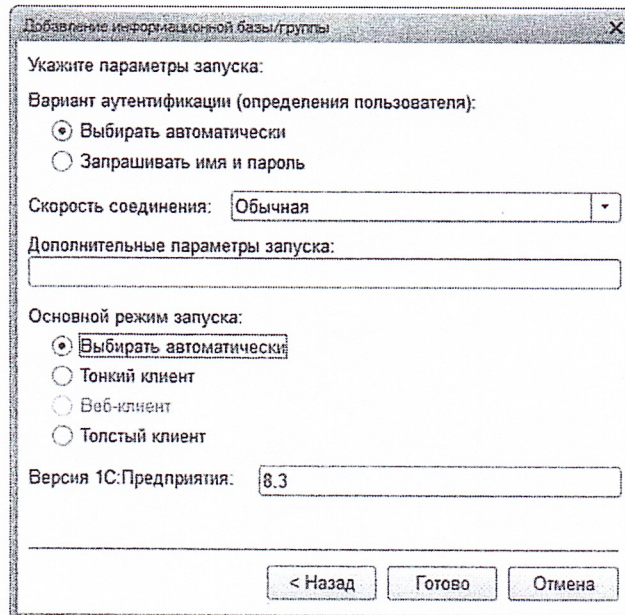
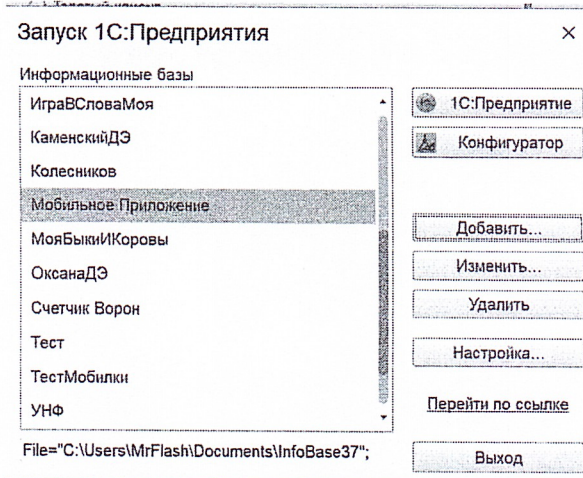


Рисунок 7. Типы запуска

В окне выбора конфигураций выбираем созданную нами базу и запускаем ее в режиме «Конфигуратор»



Выбираем пункт меню «Администрирование – Загрузить информационную базу»:

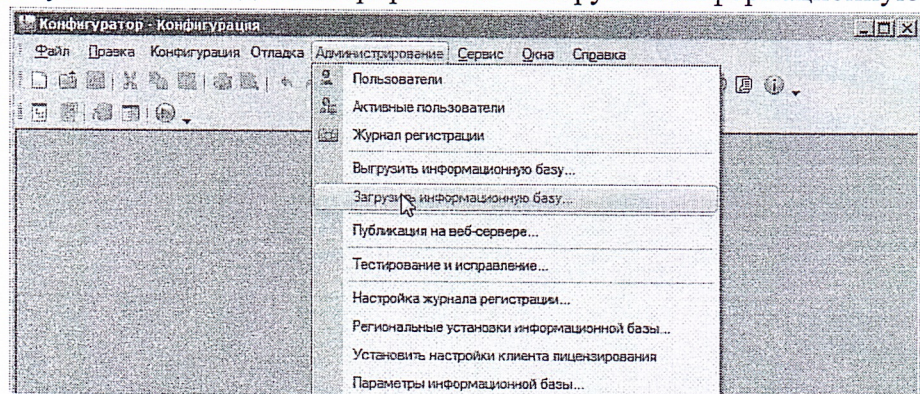


Рисунок 8. Загрузка конфигурации

Далее выбираем для загрузки файл в формате *.dt:

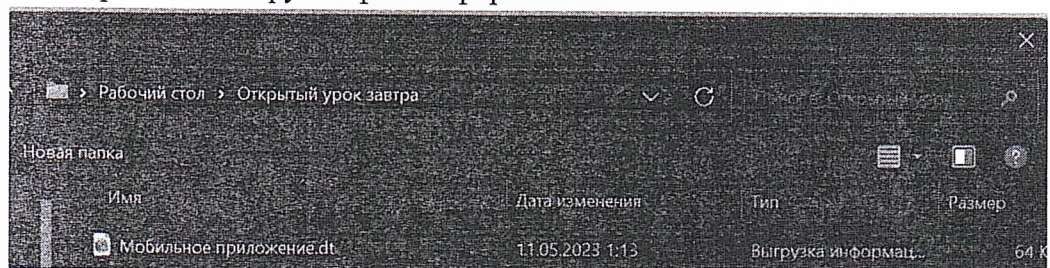


Рисунок 9. Выбор конфигурации для загрузки

На открывшееся сообщение нажимаем «Да»:

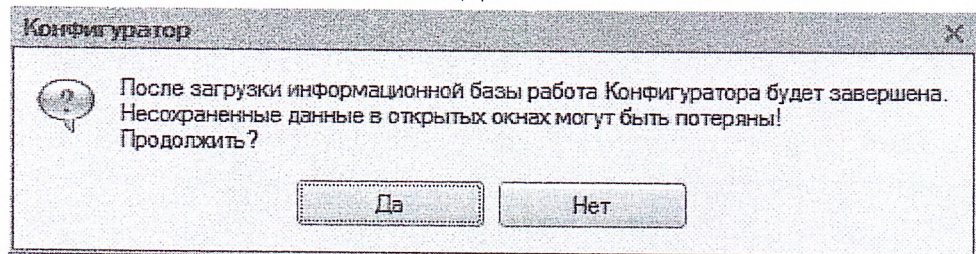


Рисунок 10. Сообщение о загрузке.

Далее нам необходимо проверить настройки конфигурации. Для этого нажмите левой кнопкой мыши на иконку конфигурации в верхней командной панели.



Рисунок 11. Иконка конфигурации
Произойдет открытие основного дерева объектов нашей конфигурации.

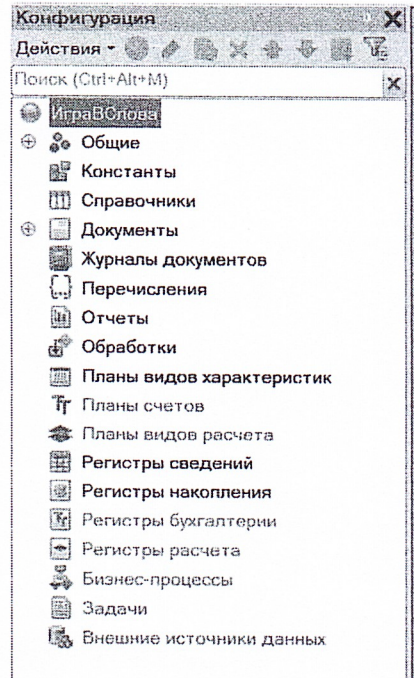


Рисунок 12. Дерево объектов конфигурации

Нажмите правой кнопкой мыши на название нашей конфигурации «Игра в слова», справа произойдет открытие формы свойств конфигурации, в пункте назначение использования убедитесь, что выбрано «Приложение для мобильной платформы»

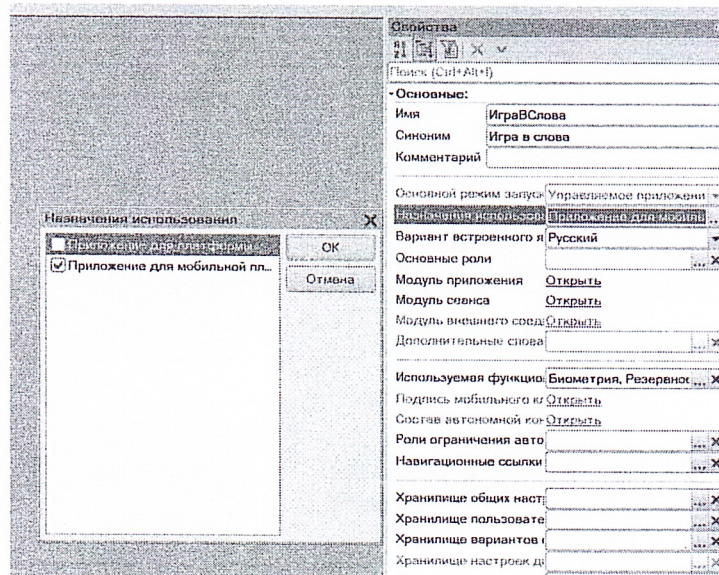


Рисунок 13. Окно свойств конфигурации.

Далее на командной панели рядом с желтым кружком располагается кнопка выбора в виде стрелочки вниз, нажмите ее, затем перейдите в пункт меню «Добавить или удалить кнопки» и выберите кнопку мобильное приложение начать отладку.

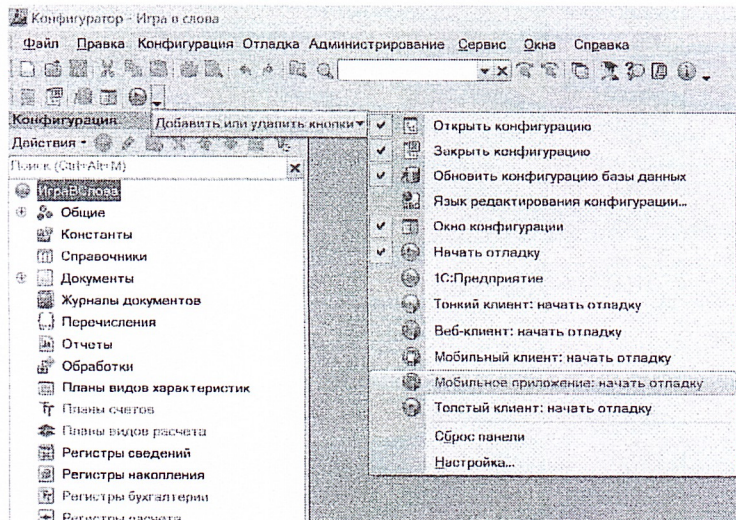


Рисунок 14. Добавление кнопки отладки в панель управления.

Теперь у нас все готово к исправлению ошибок в нашем приложении.

Первым делом нам необходимо опубликовать наше мобильное приложение для выполнения его отладки. Для этого перейдите в пункт меню «Конфигурация» затем «Мобильное приложение» и нажмите кнопку опубликовать.

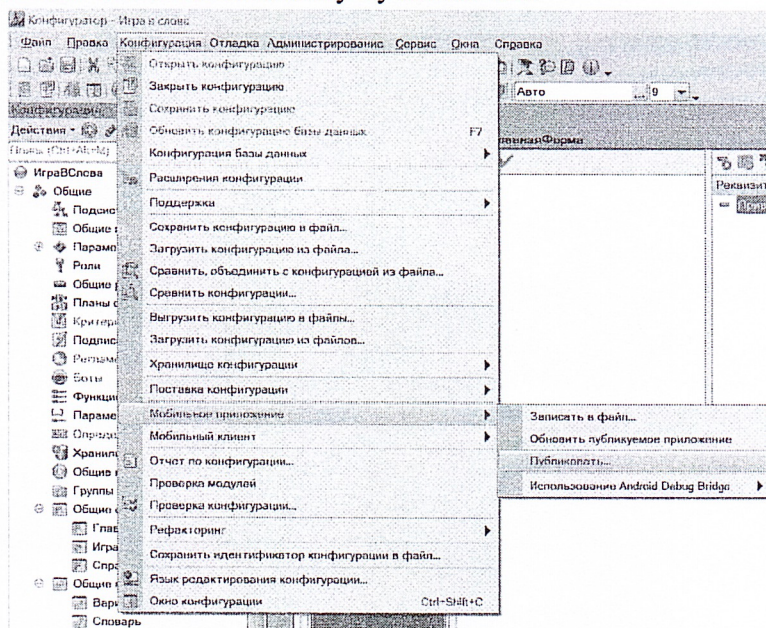


Рисунок 15. Публикация мобильного приложения

В появившемся окне оставьте все параметры по умолчанию и нажмите кнопку «Опубликовать».

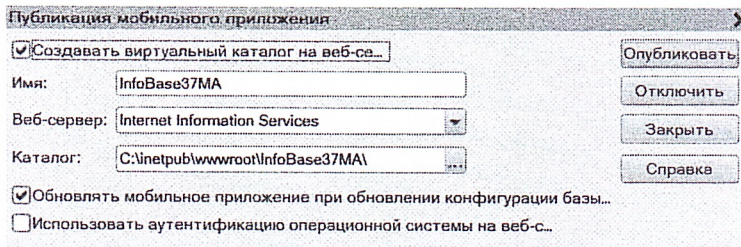


Рисунок 16. Опубликовать мобильное приложение

Раскройте пункт меню конфигурации «Общие». Здесь мы увидим, что в нашей конфигурации располагаются объекты: Такие как общие формы и общие макеты. Именно в общих формах у нас отображены основные компоненты нашего приложения.

Перед началом исправления приложения нам нужно узнать какие ошибки влияют на его работу. Для этого нажмите на кнопку «Мобильное приложение начать отладку» в верхней командной панели.

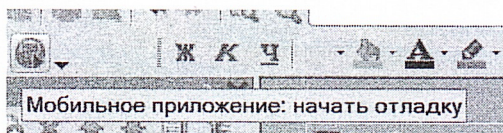


Рисунок 17. Начать отладку

Произойдет запуск эмулятора, в котором будет автоматически размещено наше приложение.

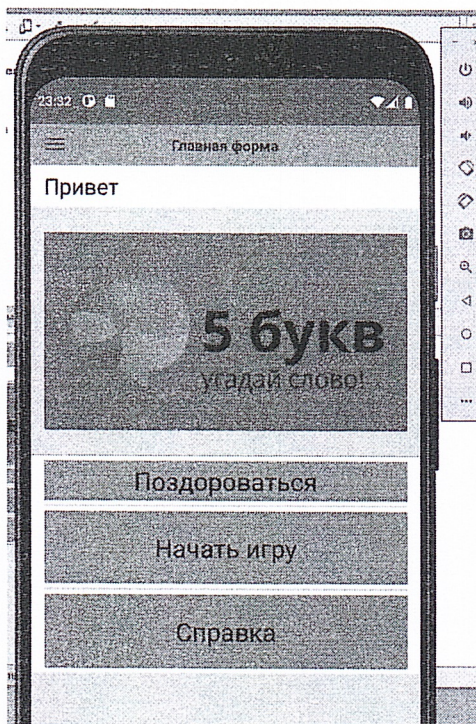


Рисунок 18. Внешний вид окна эмулятора

Для проверки работоспособности мы должны выполнить все возможные действия в данной форме.

Нажмем кнопку «Поздороваться» нам предложат ввести имя выполним это и нажмем кнопку «ОК».

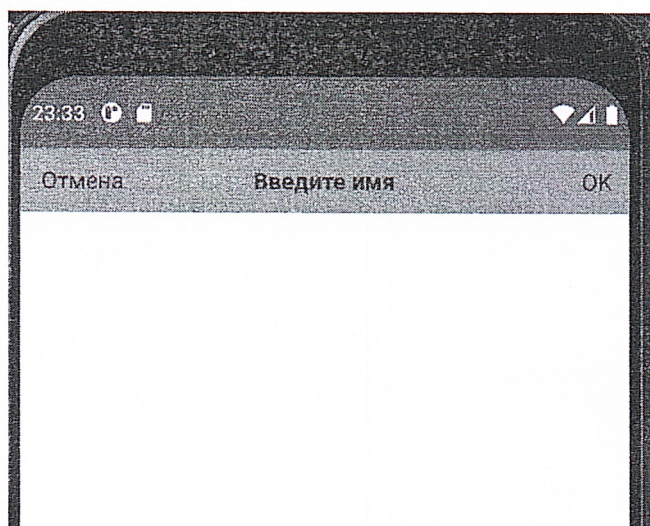


Рисунок 19. Форма ввода имени.

Главная форма приложения изменилась ошибок не возникло.

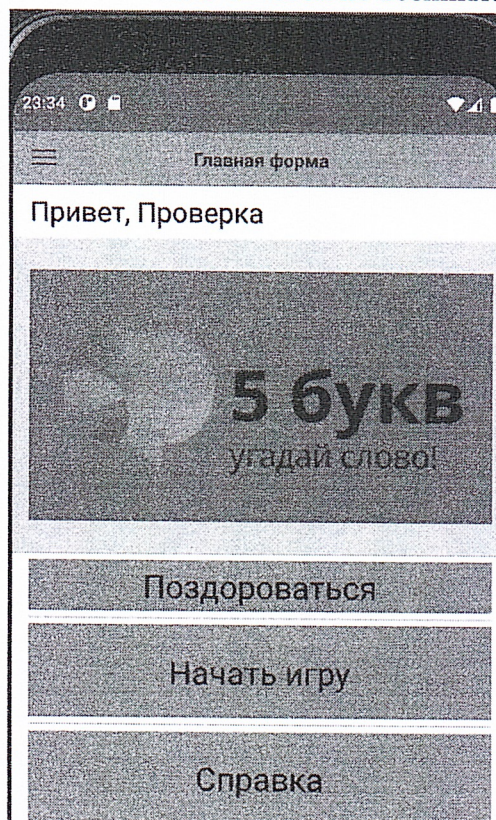


Рисунок 20. Главная форма приложения.

Затем проверим работоспособность кнопки «Справка», для этого нажмем на нее. Произошло открытие формы содержащей правила игры.

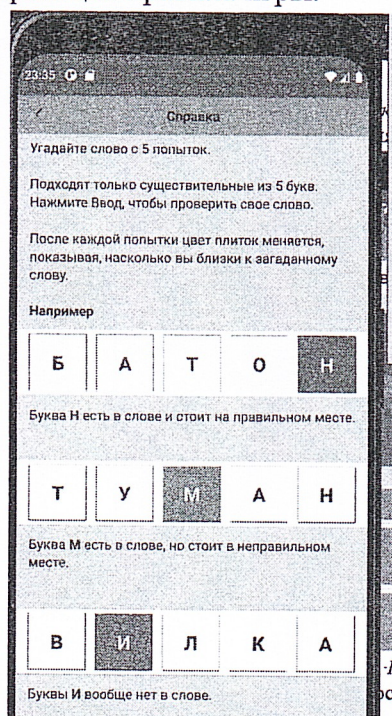


Рисунок 21. Справка.

Ошибок так же не возникло. Вернемся назад путем нажатия стрелочки в верхнем левом углу. Нажмем кнопку «Начать игру».

На главной форме возникло сообщение с ошибкой нажмем кнопку «Подробнее»

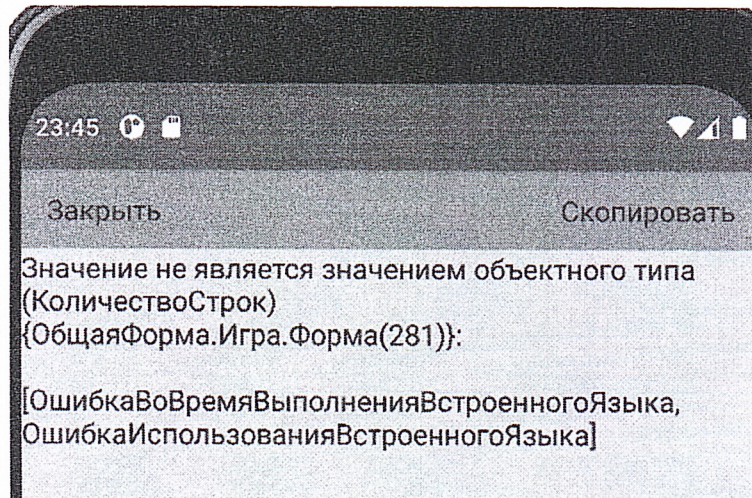


Рисунок 22. Описание ошибки.

В данном окне мы видим описание возникшей ошибки, для ее исправление вернемся в конфигурацию, раскроем вкладку «Общие» в дереве объектов конфигурации, выберем пункт «Общие формы» и в раскрывшемся списке дважды кликнем левой кнопкой мыши на объекте с названием «Игра», откроется форма объекта, на которой мы можем увидеть элементы данной формы, но в текущий момент нас интересует программный код данного объекта.

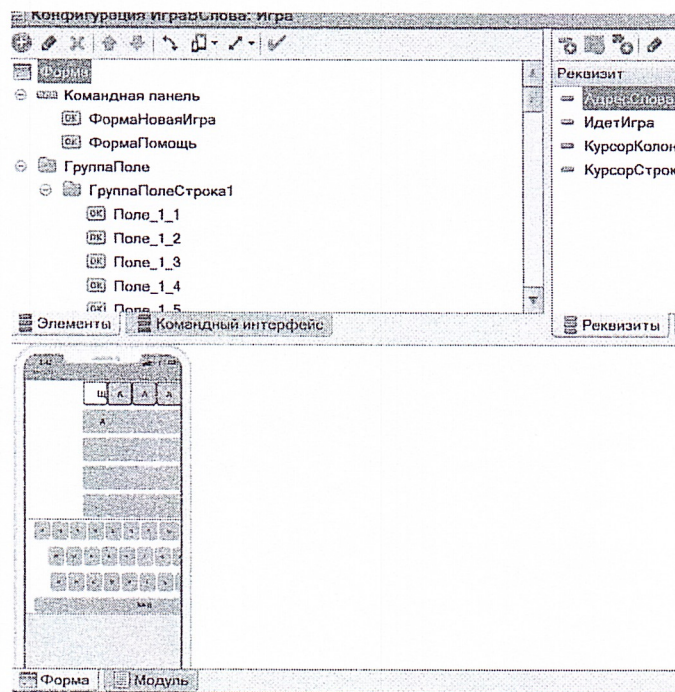


Рисунок 23. Форма объекта "Игра".

Для просмотра кода нажмите вкладку «Модуль» в левом нижнем углу раскрывшейся формы. Произойдет открытие модуля объекта «Игра» который содержит программный код.

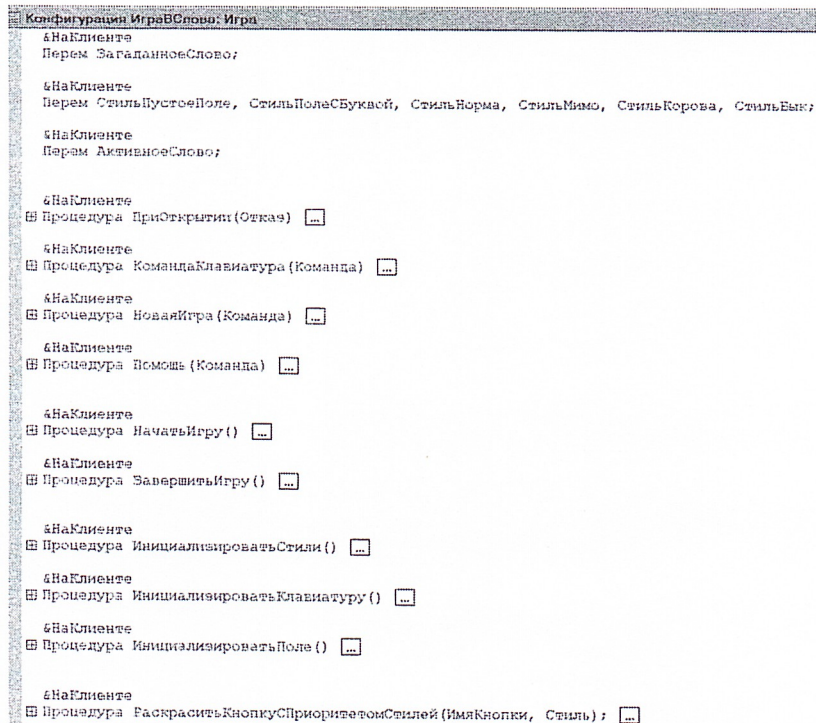


Рисунок 24. Код объекта "Игра".

Если мы обратим внимание на ошибку в эмуляторе то там указано что:
Значение не является значением объектного типа (КоличествоСтрок)
{ОбщаяФорма.Игра.Форма(281)}:

[ОшибкаВоВремяВыполненияВстроенногоЯзыка,
ОшибкаИспользованияВстроенногоЯзыка]

Данная ошибка означает неверное описание и определение функций взаимодействия мобильного приложения. Для устранения возникшей ошибки в модуле формы «Игра» в верхней командной панели, найдите поле с лупой которое служит для поиска данных в коде объекта.

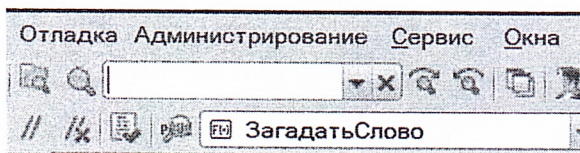


Рисунок 25. Поле поиска.

В указанном поле введите слово «КоличествоСтрок», именно на это значение возникла ошибка в нашей конфигурации и нажмите клавишу «Enter».

Платформа найдет искомое слово в коде и укажет на него указателем.

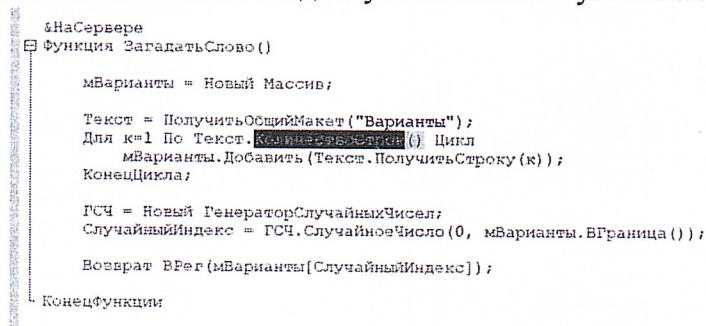


Рисунок 26. Найденное слово.

Предыдущий исполнитель допустил ряд ошибок которые нам предстоит исправить, он не учел, что мобильное приложение 1С не работает с текстовыми файлами на прямую и в объектах конфигурации создал два общих текстовых макета: «Варианты» и «Справка», также он использует функции работы с текстовыми документами, такие как: «КоличествоСтрок», «ПолучитьСтроку». Для исправления ситуации нам необходимо создать Общий макет, с типом «Двоичные данные», и произвести исправление кода. Для этого в дереве объектов конфигурации раскройте вкладку «Общие макеты» нажмите по данному названию правой кнопкой мыши и выберите пункт «Добавить». Произойдет открытие «Конструктора макета»

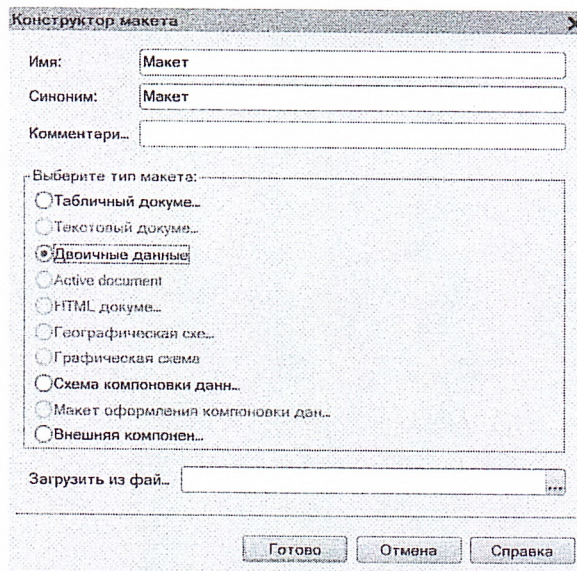


Рисунок 27. Конструктор макета.

В поле имя введите имя макета например: «Вариант» в типе макета выберите двоичные данные и нажмите «Готово». Откроется форма загрузки двоичных данных, закройте ее на время. Нам необходимо внести в новый макет данные из объекта «Варианты». Для этого откройте объект «Варианты» путем двойного нажатия левой кнопкой мыши, откроется окно с данными которые содержатся в объекте, указанные данные необходимо полностью выделить путем нажатия сочетаний клавиш Ctrl + A и скопировать выделенный текст.

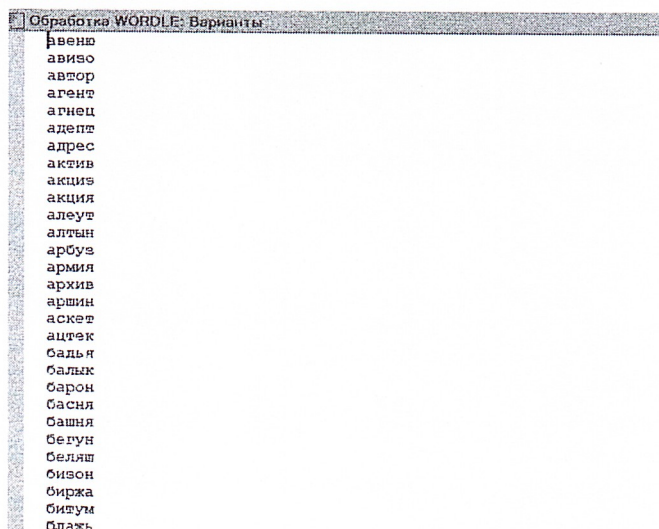


Рисунок 28. Данные объекта Варианты.

Выделенный текст необходимо записать в блокнот в формате .txt. Для этого на рабочем столе в любом свободном участке, нажмите правой кнопкой мыши в появившемся меню выберите: Создать – Таксовый документ. Задайте ему имя, например Варианты, откройте его и вставьте скопированные данные. Затем закройте блокнот и сохраните изменения.

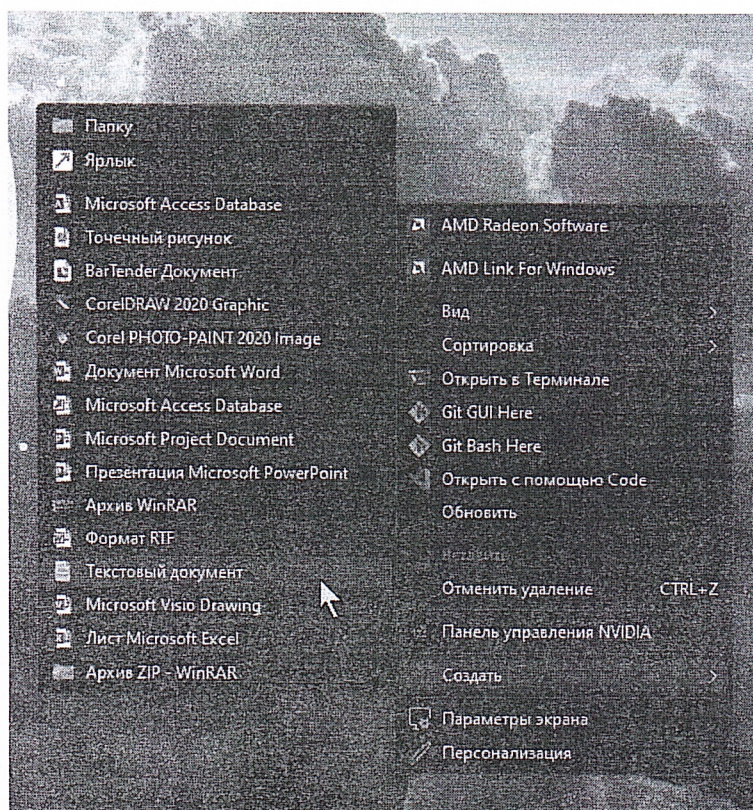


Рисунок 29. Создание текстового документа.

Вернитесь в конфигурацию, откройте новый макет с именем «Вариант», выберите пункт меню «Загрузить из файла» и выберите созданный нами текстовый документ «Вариант», нажмите кнопку «Ок». Удалите Общий макет «Варианты» и задайте новому макету его имя. Перейдем к редактированию кода.

Мы должны обратиться к новому общему макету, получить содержащиеся в нем данные и записать их в массив.

В нашем случае код должен выглядеть следующим образом.

```

&НаСервере
Функция ЗагадатьСлово()

    // Создаем пустой массив
    мВарианты = Новый Массив;

    // Получаем общий макет "Варианты" на мобильном клиенте
    Текст = ПолучитьОбщийМакет("Варианты");

    // Открываем поток для чтения двоичных данных
    Поток = Текст.ОткрытьПотокДляЧтения();
    // Создаем объект для чтения текста из потока, указываем кодировку UTF8
    Чтение = Новый ЧтениеТекста(Поток, КодировкаТекста.UTF8);
    // Считываем все данные из потока и сохраняем в переменную Результат
    Результат = Чтение.Прочитать();
    // Закрываем объект чтения текста и поток
    Чтение.Закрыть();
    Поток.Закрыть();

    // Заполняем массив мВарианты данными из переменной Результат
    Для к=1 По СтрЧислоСтрок(Результат) Цикл
        мВарианты.Добавить(СтрПолучитьСтроку(Результат,к));
    КонецЦикла;
    // Создаем объект для генерации случайных чисел
    ГСЧ = Новый ГенераторСлучайныхЧисел;

    // Генерируем случайный индекс для выбора случайного значения из массива
    СлучайныйИндекс = ГСЧ.СлучайноеЧисло(0, мВарианты.ВГраница());

    // Возвращаем случайное значение из массива, приведенное к верхнему регистру
    Возврат ВРег(мВарианты[СлучайныйИндекс]);

КонецФункции

```

Рисунок 30. Исправленный код.

Код текстом:

&НаСервере

Функция ЗагадатьСлово()

 // Создаем пустой массив

 мВарианты = Новый Массив;

 // Получаем общий макет "Варианты" на мобильном клиенте

 Текст = ПолучитьОбщийМакет("Варианты");

 // Открываем поток для чтения двоичных данных

 Поток = Текст.ОткрытьПотокДляЧтения();

 // Создаем объект для чтения текста из потока, указываем кодировку UTF8

 Чтение = Новый ЧтениеТекста(Поток, КодировкаТекста.UTF8);

 // Считываем все данные из потока и сохраняем в переменную Результат

 Результат = Чтение.Прочитать();

 // Закрываем объект чтения текста и поток

 Чтение.Закрыть();

 Поток.Закрыть();

 // Заполняем массив мВарианты данными из переменной Результат

 Для к=1 По СтрЧислоСтрок(Результат) Цикл

 мВарианты.Добавить(СтрПолучитьСтроку(Результат,к));

 КонецЦикла;

 // Создаем объект для генерации случайных чисел

 ГСЧ = Новый ГенераторСлучайныхЧисел;

 // Генерируем случайный индекс для выбора случайного значения из массива

 СлучайныйИндекс = ГСЧ.СлучайноеЧисло(0, мВарианты.ВГраница());

 // Возвращаем случайное значение из массива, приведенное к верхнему регистру

 Возврат ВРег(мВарианты[СлучайныйИндекс]);

КонецФункции

После проделанных манипуляций обновите конфигурацию путем нажатия данной кнопки на командной панели, и запустите отладку мобильного приложения, нажмите кнопку «Начать игру», на данный момент ошибки устранены, появляется основная форма приложения, с которым можно взаимодействовать, попробуем ввести слово и нажмем кнопку «Ввод». Возникла новая ошибка нажмем кнопку «Подробнее».

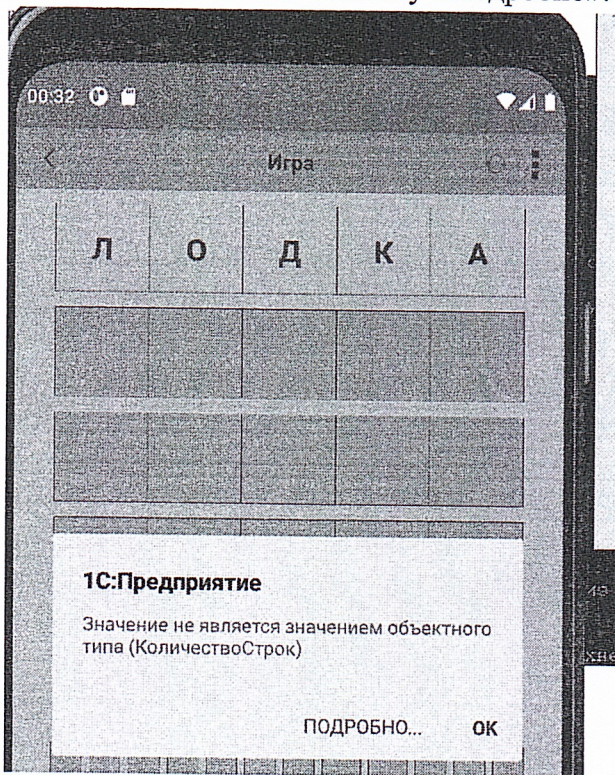


Рисунок 31. Основная форма приложения.

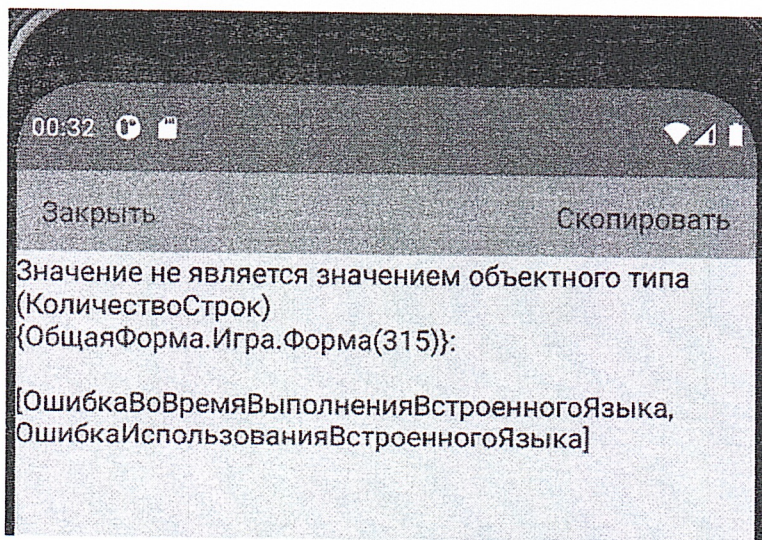


Рисунок 32. Текст ошибки.

Ошибка схожа с предыдущей, нам необходимо выполнить те же действия, найти строку ошибки в коде конфигурации в данном случае ошибка содержится в процедуре «ЗаполнитьСловарь»

```

&НаСервере
□ Процедура ЗаполнитьСловарь ()
    соСловарь = Новый Соответствие;

    Текст = ПолучитьОбщийМакет("Словарь");
    Для к=1 По Текст.КоличествоСтрок() Цикл
        соСловарь.Вставить(Текст.ПолучитьСтроку(к), Истина);
    КонiecЦикла;

    АдресСловаря = ПоместитьВоВременноеХранилище(соСловарь, УникальныйИдентификатор);
КонецПроцедуры

```

Рисунок 33. Процедура "ЗаполнитьСловарь"

Нам необходимо произвести те же действия с общим макетом что и выше, только на этот раз Общий макет называется «Словарь». После проделанных манипуляций нам необходимо исправить код процедуры, он будет выглядеть следующим образом:

```

&НаСервере
□ Процедура ЗаполнитьСловарь ()
    // Создаем новый объект типа Соответствие
    соСловарь = Новый Соответствие;

    // Получаем общий макет "Словарь" на мобильном клиенте
    Текст = ПолучитьОбщийМакет("Словарь");

    // Открываем поток для чтения двоичных данных
    Поток = Текст.ОткрытьПотокДляЧтения();

    // Создаем объект для чтения текста из потока, указываем кодировку UTF8
    Чтение = Новый ЧтениеТекста(Поток, КодировкаТекста.UTF8);

    // Считываем все данные из потока и сохраняем в переменную Результат
    Результат = Чтение.Прочитать();

    // Закрываем объект чтения текста и поток
    Чтение.Закреть();
    Поток.Закреть();

    // Заполняем соСловарь парами "ключ-значение" из переменной Результат
    Для к=1 По СтрЧислоСтрок(Результат) Цикл
        соСловарь.Вставить(СтрПолучитьСтроку(Результат, к), Истина);
    КонiecЦикла;

    // Помещаем объект соСловарь во временное хранилище с уникальным идентификатором
    АдресСловаря = ПоместитьВоВременноеХранилище(соСловарь, УникальныйИдентификатор);
КонецПроцедуры

```

Рисунок 34. Код процедуры ЗаполнитьСловарь после исправления

Листинг кода:

&НаСервере

Процедура ЗаполнитьСловарь()

```

// Создаем новый объект типа Соответствие
соСловарь = Новый Соответствие;
// Получаем общий макет "Словарь" на мобильном клиенте
Текст = ПолучитьОбщийМакет("Словарь");
// Открываем поток для чтения двоичных данных
Поток = Текст.ОткрытьПотокДляЧтения();
// Создаем объект для чтения текста из потока, указываем кодировку UTF8
Чтение = Новый ЧтениеТекста(Поток, КодировкаТекста.UTF8);
// Считываем все данные из потока и сохраняем в переменную Результат
Результат = Чтение.Прочитать();
// Закрываем объект чтения текста и поток
Чтение.Закреть();
Поток.Закреть();

```

```

// Заполняем соСловарь парами "ключ-значение" из переменной Результат
Для к=1 По СтрЧислоСтрок(Результат) Цикл
    соСловарь.Вставить(СтрПолучитьСтроку(Результат,к), Истина);
КонецЦикла;
// Помещаем объект соСловарь во временное хранилище с уникальным
идентификатором
    АдресСловаря = ПоместитьВоВременноеХранилище(соСловарь,
УникальныйИдентификатор);
КонецПроцедуры

```

После проведенных исправлений, обновляем конфигурацию нашего приложения и заново запускаем отладку.

После начала игры и ввода первого слова. Приложение работает в штатном режиме и выполняет свой функционал



Рисунок 35. Штатная работа приложения.