



**КВАНТОРИУМ**  
КАЛИНИНГРАД

Проект вводного модуля  
01.03.2018 г. – 29.05.2018 г.  
направления  
**IT-КВАНТУМ**



Слепухин Даниил

Аверьянов Артём

Черненко Слава



# ПРОБЛЕМА И АУДИТОРИЯ

**Проблема:** Некоторым ученикам требуется много времени для заучивания стихов

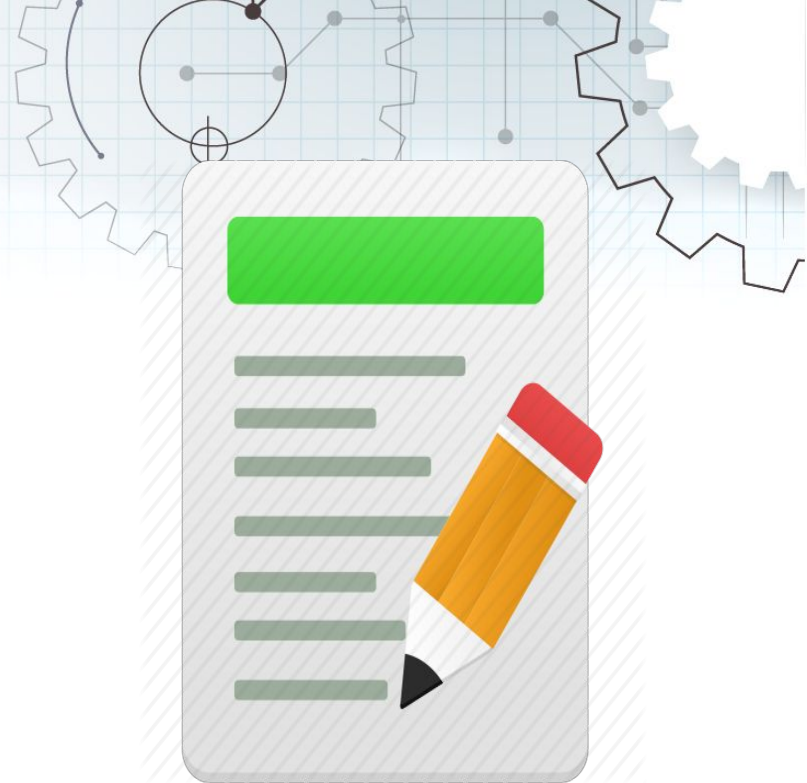
**Аудитория:** Учащихся образовательных учреждений.

Исследование аналогов показало: что точно таких приложений, как у нас, нет  
В основном, всё что у нас есть, находится в разных приложениях, что показывает – наше приложение уникально.





**КВАНТОРИУМ**  
КАЛИНИНГРАД



# ITvoice

---

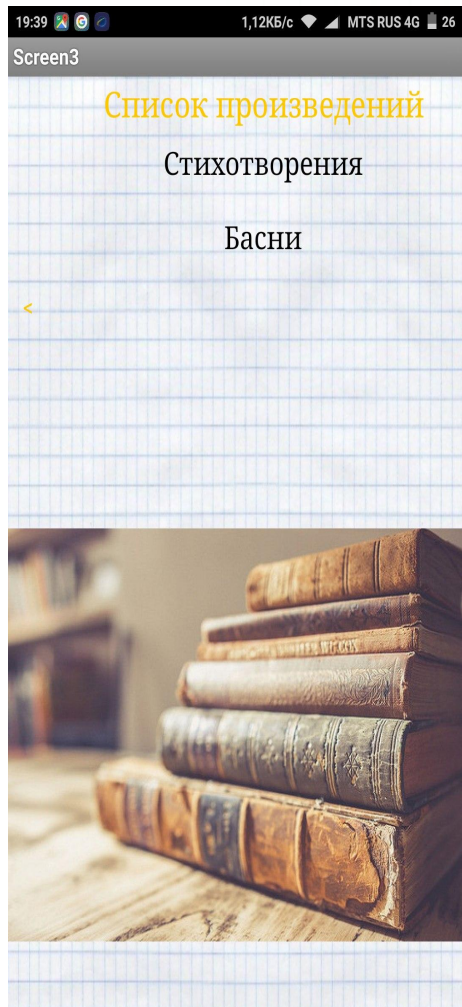
Мобильное приложение, помогающее детям учить стихи





# ОПИСАНИЕ ПРОЕКТА

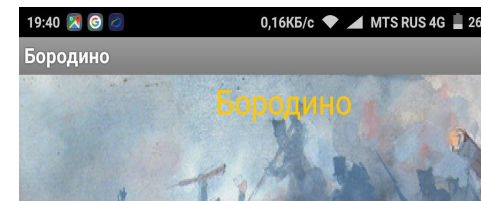
Наше приложение сделано на ОС Android в среде “Mit app inventor”



```
код Когда? Цикл  
зациклить цикл РассказыватьПоле1.ПолучитьТекст
```

```
код РассказыватьПоле1.ПолучитьТекст  
результат  
зациклить цикл  
1. получить Набрись1.Текст1 РассказыватьПоле1.Результат1  
2. когда РассказыватьПоле1.Результат1.0.1 -- Слова дари мир не даром Москва столица  
3. получить Набрись2.Текст1 Верно  
4. когда получить Набрись2.Текст1 Неверно
```

```
код Когда? Цикл  
зациклить цикл КогдаДрайЖанКаминФорме.Борозды
```



Говори



# ПЕРСПЕКТИВЫ РАЗВИТИЯ ПРОЕКТА

В будущем, мы планируем:

- добавить больше стихотворений и басень ,
- реализовать подачу для всех в понятной форме.

```
383
384     endless_query = self.
385     endless_query.start_position = self.start_position
386     self._all_patches = self._generate_patches()
387     return self._all_patches
388
389     def compute_percentile(self, percentage):
390         """
391         Returns a Position object that represents percentage%-far-of-the-way
392         through the larger task, as specified by this query.
393
394         @param percentage a number between 0 and 100.
395         """
396         all_patches = self._generate_patches()
397         return all_patches[int(len(all_patches) * percentage)]
398
399     def generate_patches(self):
400         """
401         Generates a list of Position objects that satisfy the given conditions given
402         the root directory and the path filter.
403
404         @param root_directory the root directory
405         @param path_filter a function that filters patches
406         """
407         root_directory = self.root_directory
408         path_filter = self.path_filter
409         all_patches = self._generate_patches(root_directory, path_filter)
410         return all_patches
```

