

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**Национальный исследовательский университет Высшая Школа Экономики**  
Факультет компьютерных наук  
Департамент программной инженерии  
Образовательная программа «Программная инженерия»

**СОГЛАСОВАНО**  
Научный руководитель, приглашенный  
преподаватель департамента  
программной инженерии

**УТВЕРЖДАЮ**  
Академический руководитель  
образовательной программы  
«Программная инженерия»

\_\_\_\_\_  
Сосновский Г. М.  
«2» Апреля 2024 г.

\_\_\_\_\_  
Павлочев Н. А.  
«2» Апреля 2024 г.

**iOS библиотека с успокаивающими UI элементами**  
**Пояснительная записка**  
**ЛИСТ УТВЕРЖДЕНИЯ**  
**RU.17701729.05.01-01 ТЗ 01-1-ЛУ**

**Исполнитель:**

студент группы БПИ224

\_\_\_\_\_  
/ Безруков Д. Р. /  
«2» Апреля 2024 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	RU.17701729.05.01-01 ТЗ 01-1 01-1

**Москва 2024**

**УТВЕРЖДЕН**  
**RU.17701729.05.01-01 ТЗ 01-1 01-1-ЛЮ**

<i>Инв. № подл</i>	<i>Подп. и дата</i>	<i>Взам. инв. №</i>	<i>Инв. № дубл.</i>	<i>Подп. и дата</i>
RU.17701729.05.01-01 ТЗ 01-1 01-1-ЛЮ				

**iOS библиотека с успокаивающими UI элементами**  
**Пояснительная записка**  
**RU.17701729.05.01-01 ТЗ 01-1 01-1-ЛЮ**  
**Листов 18**

## СОДЕРЖАНИЕ

<b>1</b>	<b>ВВЕДЕНИЕ .....</b>	<b>4</b>
1.1	НАИМЕНОВАНИЕ ПРОГРАММЫ .....	4
<b>2</b>	<b>НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ.....</b>	<b>5</b>
2.1	НАЗНАЧЕНИЕ ПРОГРАММЫ .....	5
2.1.1	Функциональное назначение .....	5
2.1.2	Эксплуатационное назначение .....	5
2.2	КРАТКАЯ ХАРАКТЕРИСТИКА ОБЛАСТИ ПРИМЕНЕНИЯ.....	5
<b>3</b>	<b>ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ .....</b>	<b>6</b>
3.1	ПОСТАНОВКА ЗАДАЧИ НА РАЗРАБОТКУ ПРОГРАММЫ.....	6
3.2	ОПИСАНИЕ И ОБОСНОВАНИЕ АЛГОРИТМА РАБОТЫ ПРОГРАММЫ .....	7
3.2.1	Общий алгоритм работы программы .....	7
3.2.2	Выбор архитектуры проекта.....	8
3.2.3	Внедрение SwiftPM.....	8
3.2.4	Алгоритм работы отдельных компонентов .....	9
3.2.5	Продумывание написания логики модулей.....	9
3.3	ОПИСАНИЕ И ОБОСНОВАНИЕ ВЫБОРА СПОСОБА ОРГАНИЗАЦИИ ВХОДНЫХ И ВЫХОДНЫХ ДАННЫХ.....	9
3.3.1	Описание метода организации входных данных .....	9
3.3.2	Обоснование метода организации входных данных .....	9
3.3.3	Описание метода организации выходных данных .....	10
3.3.4	Обоснование метода организации выходных данных .....	10
3.4	ОПИСАНИЕ И ОБОСНОВАНИЕ ВЫБОРА СОСТАВА ТЕХНИЧЕСКИХ И ПРОГРАММНЫХ СРЕДСТВ .....	10
3.4.1	Состав технических и программных средств .....	10
3.4.2	Обоснование выбора технических и программных средств.....	10
<b>4</b>	<b>ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ .....</b>	<b>11</b>
4.1	ОРИЕНТИРОВОЧНАЯ ЭКОНОМИЧЕСКАЯ ЭФФЕКТИВНОСТЬ.....	11
4.2	ПРЕДПОЛАГАЕМАЯ ПОТРЕБНОСТЬ .....	11
4.3	ЭКОНОМИЧЕСКИЕ ПРЕИМУЩЕСТВА РАЗРАБОТКИ ПО СРАВНЕНИЮ С ОТЕЧЕСТВЕННЫМИ И ЗАРУБЕЖНЫМИ АНАЛОГАМИ .....	11
	<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....</b>	<b>14</b>
	<b>ПРИЛОЖЕНИЕ 1 .....</b>	<b>16</b>
	<b>ПРИЛОЖЕНИЕ 2 .....</b>	<b>17</b>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 ТЗ 01-1				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

## АННОТАЦИЯ

Данный программный документ представляет собой пояснительную записку к программному проекту «iOS библиотека с успокаивающими UI элементами».

Раздел «Введение» включает в себя наименование программы и документ, на основании которого ведётся разработка, с указанием организации, утвердившей данный документ.

В разделе «Назначение и область применения» содержатся функциональное и эксплуатационное назначение программы и краткая характеристика области её применения.

В разделе «Технические характеристики» присутствуют следующие подразделы: постановка задачи на разработку программы, описание функционирования программы, описание и обоснование алгоритма работы программы, описание и обоснование выбора метода организации входных и выходных данных, описание работы с базой данных, описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами.

Программный документ разработан в соответствии с требованиями:

1. ГОСТ 19.101-77 Виды программ и программных документов [1];
2. ГОСТ 19.102-77 Стадии разработки [2];
3. ГОСТ 19.103-77 Обозначения программ и программных документов [3];
4. ГОСТ 19.104-78 Основные надписи [4];
5. ГОСТ 19.105-78 Общие требования к программным документам [5];
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [7].

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

Перед прочтением данного документа следует ознакомиться с терминологией, приведенной в Приложении 1.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 ТЗ 01-1				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

## 1 ВВЕДЕНИЕ

### 1.1 Наименование программы

**Наименование темы разработки:** «iOS библиотека с успокаивающими UI элементами».

**Наименование темы разработки на английском языке:** «iOS Package with Fidget UI Elements».

**Условное обозначение темы разработки:** «iOS-библиотека “ZenUI”».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 ТЗ 01-1				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

## 2 НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

### 2.1 Назначение программы

#### 2.1.1 Функциональное назначение

Разрабатываемая iOS-библиотека и приложение для демонстрации «ZenUI» предназначены для предоставления множества способов бесцельного взаимодействия с элементами интерфейса, таких как кнопки, различные игры и элементы дизайна, которые могут или вибрировать, или издавать звуки, или способствовать расслаблению и отвлечению пользователя.

#### 2.1.2 Эксплуатационное назначение

iOS-библиотека «ZenUI» предоставляет разработчикам возможности интеграции успокаивающих и развлекательных элементов в их приложения на платформе iOS. Основными конечными потребителями разрабатываемой библиотеки являются разработчики приложений, но приложение, которое будет демонстрировать все функции данной библиотеки, также подходит для обычных пользователей. Данное приложение предусмотрено к использованию на смартфонах с операционной системой iOS 15.0 или выше. Для работы с приложениями не требуются специальные навыки, необходимы лишь базовые для использования мобильного устройства.

### 2.2 Краткая характеристика области применения

ZenUI — это iOS-библиотека, разработанная для внедрения в приложения, где разработчики хотят создать легкое, развлекательное и успокаивающее взаимодействие с пользователем. Данная библиотека включает в себя различные элементы такие, как модели известных элементов дизайна, цветовые схемы и звуковые эффекты.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 ТЗ 01-1				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

### 3 ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

#### 3.1 Постановка задачи на разработку программы

Согласно требованиям к функциональным характеристикам, реализован следующий функционал:

##### 3.1.1 Основное

3.1.1.1 Возможность интерактивно взаимодействовать с UI-отображениями реализованных элементов в приложении.

##### 3.1.2 Навигация

3.1.2.1 Возможность перемещаться между разделами “Главная” и выбранным UI-элементом.

##### 3.1.3 “Главная”

3.1.3.1 Возможность взаимодействовать с реализациями UI-элементов:

3.1.3.1.1 Возможность взаимодействовать с кнопкой-переключателем.

3.1.3.1.2 Возможность взаимодействовать с “Колыбелью Ньютона”.

3.1.3.1.3 Возможность взаимодействовать с имитацией “Резиновой уточки”.

3.1.3.1.4 Возможность играть в карточную игру на память “Concentration”.

3.1.3.1.5 Возможность взаимодействовать со спиннером.

3.1.3.1.6 Возможность взаимодействовать с приложением типа “Кликер”.

3.1.3.1.7 Возможность “подбрасывать” монетку.

3.1.3.1.8 Возможность взаимодействовать с шаром предсказаний.

3.1.3.1.9 Возможность взаимодействовать с игровой костью.

3.1.3.1.10 Возможность взаимодействовать с генератором случайных чисел.

3.1.3.1.11 Возможность взаимодействовать с генератором случайной игровой карты.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 ТЗ 01-1				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

3.1.3.1.12 Возможность взаимодействовать с ромашкой для игры в “любит-не-любит”.

3.1.3.1.13 Возможность нажимать на имитацию кнопки.

3.1.3.1.14 Возможность взаимодействовать со слайдером.

3.1.3.1.15 Возможность взаимодействовать с ксилофоном.

3.1.3.1.16 Возможность взаимодействовать с доской для рисования.

3.1.3.1.17 Возможность играть в крестики-нолики.

3.1.3.1.18 Возможность играть в “Камень, ножницы, бумага”.

3.1.3.1.19 Возможность играть в “2048”.

3.1.3.1.20 Возможность играть в игру “Четыре в ряд”.

### 3.1.4 “Настройки”

3.1.4.1 В разделе “Главное” возможность изменить параметры звука.

3.1.4.2 В разделе конкретного UI-элемента возможность выйти в раздел Главного меню.

## 3.2 Описание и обоснование алгоритма работы программы

### 3.2.1 Общий алгоритм работы программы

Библиотека и iOS приложение для демонстрации функционала были написаны на языке программирования Swift. Базовый функционал библиотеки и экраны представлены ниже (рис. 1).

Главное меню написано с помощью современного и удобного для поставленной задачи инструмента SwiftUI. Данный фреймворк был заранее определен техническим заданием. Однако его выбор также можно обосновать тем, что он идеально подходит программисту при создании приложения на Swift, когда ставится цель реализовать сетку иконок, а также продумать логику навигации между экранами. Решение не внедрять дополнительно библиотеку UIKit фреймворка Cocoa Touch при создании главного экрана обосновано простотой логики такого типа приложения, как демонстрация существующего функционала.

Все модули, которые реализовывают логику успокаивающих UI-элементов, находятся в отдельной библиотеке ZenUILibrary, которая была реализована с помощью Swift Package Manager. Зависимости между приложением и пакетом были

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 ТЗ 01-1				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и



заранее установлены, так же, как и соответствующие инициализаторы в классах с модификаторами доступа `public`, чтобы избежать потенциальных ошибок при сборке.

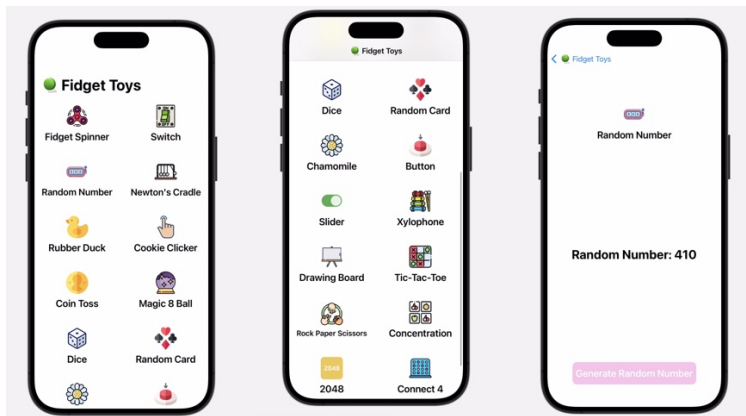


Рисунок 1. Иллюстрация первоначального дизайна приложения

### 3.2.2 Выбор архитектуры проекта

Для мобильных приложений рекомендуемые архитектуры — это MVVM+C, VIPER, MVP, MVC. Однако из-за стека технологий в виде SwiftUI и SwiftPM мой выбор упал на MVVM из-за его популярности при использовании SwiftUI. Также важным фактом является то, что из-за SwiftUI MVC невозможно использовать в силу ее плохой совместимости с данным фреймворком. Другие архитектуры не были выбраны из-за значительного роста в сложности и излишне подробной декомпозиции для небольшого проекта. И поэтому конечная архитектура приложения в основном напоминает архитектуру приложения MVVM. За исключением того, что логика UI элементов и модель FidgetToy, описывающаяся структурой и необходимая для реализации Fidget UI-элементов, были вынесены в пакет ZenUILibrary.

### 3.2.3 Внедрение SwiftPM

В изначальных требованиях было сказано воспользоваться SPM для разработки пакета модулей и удобной установки зависимостей. И соответственно я выбрал данный способ. SPM появился совсем недавно, но уже начал завоевывать популярность, отвоевывая ее от самого распространенного на данный момент CocoaPods. Немалым фактором стало и то, что это официальный менеджер зависимостей от Apple. При изучении я столкнулся с рядом достоинств и недостатков. Главный плюс — это то, что SPM является официальным решением от Apple, и поэтому нет необходимости дополнительно что-то устанавливать. Также он позволяет избавиться от project-файлов (применяемых в CocoaPods), а вместе с ними и от множества потенциальных конфликтов. Главным минусом стало то, что при работе может внезапно потребоваться перезапустить кэш пакета, что обычно занимает 1–1.5 минуты. А также Xcode самостоятельно может внезапно решить запустить резолвинг пакетов, что негативно сказывается на работе с приложением, и из-за чего зачастую интерфейс начинает очень сильно тормозить.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 ТЗ 01-1				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

### 3.2.4 Алгоритм работы отдельных компонентов

Пользователь приложения выбирает понравившийся UI-элемент нажатием. Затем открывается новый экран с выбранным UI-элементом. Разберем на простом примере элемента дизайна – генератора случайных чисел. Пользователь нажимает на иконку и ему открывается экран, где он может нажать на появившуюся кнопку и получить случайное число, повторным нажатием выбирается новое случайное число. Чтобы выйти с этого экрана, нужно в левом верхнем углу экрана нажать на стрелочку. Логика компонентов UI посредством SPM была вынесена в отдельную библиотеку, функционал которой был протестирован в специальном приложении ZenUI.

### 3.2.5 Продумывание написания логики модулей

При создании другого UI-элемента – Fidget Spinner, я столкнулся с неожиданной проблемой, которая заключалась в том, что для простого закручивания объекта, необходимо было глубоко разобраться в построении логики этого модуля. Необходимо было разобраться, с каким ускорением это должно происходить, как должно останавливаться и как пользователь сам может остановить его. И тем самым, на удивление, такой простой элемент интерфейса позволил мне лучше разобраться в самом языке и построении логики при написании проекта.

## 3.3 Описание и обоснование выбора способа организации входных и выходных данных

### 3.3.1 Описание метода организации входных данных

Чтобы взаимодействовать с интерфейсом приложения, пользователь касается экрана, и затем данные от этого действия поступает на вход. Для выбора успокаивающего элемента интерфейса в Главном меню необходимо выполнить выбор с помощью нажатия. Для выхода из успокаивающего UI-элемента пользователь нажимает соответствующую кнопку в левом углу экрана, которая перенесет в раздел “Главное”. Для отключения звука пользователь может нажать на соответствующую иконку.

### 3.3.2 Обоснование метода организации входных данных

Этот способ организации входных данных является наиболее простым и удобным для мобильного приложения, а также соответствует всем требованиям, предусмотренным в техническом задании.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 ТЗ 01-1				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

### 3.3.3 Описание метода организации выходных данных

На экране пользователь, взаимодействуя с UI-элементом или игрой, получает на выход или текст, или звук, или визуальное отображение произведенных действий.

### 3.3.4 Обоснование метода организации выходных данных

Этот способ организации выходных данных является наиболее простым и удобным для мобильного приложения, а также соответствует всем требованиям, предусмотренным в техническом задании.

## 3.4 Описание и обоснование выбора состава технических и программных средств

### 3.4.1 Состав технических и программных средств

- 1) Смартфоны под управлением ОС iOS версии 15.0 и выше.
- 2) Не менее 200 МБ свободного места на устройстве.

### 3.4.2 Обоснование выбора технических и программных средств

Версию iOS 15.0 поддерживает абсолютное большинство телефонов на этой операционной системе, начиная с iPhone 6s 2015 года выпуска. В силу этого факта поддержка более старых версий не является целесообразной.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 ТЗ 01-1				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

## 4 ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

### 4.1 Ориентировочная экономическая эффективность

В рамках проекта расчет экономической эффективности программного продукта не производился.

### 4.2 Предполагаемая потребность

Данный программный продукт будет интересен обычным пользователям и разработчикам мобильных приложений, желающим интегрировать элементы, способствующие расслаблению и развлечению пользователей. ZenUI предоставляет разнообразные инструменты, такие как элементы дизайна, цветовые схем и звуковые эффекты, чтобы создать приятное и спокойное взаимодействие с приложением. Это особенно ценно для разработчиков, которые хотят улучшить пользовательский опыт, сделав его более расслабляющим и приятным, что может помочь пользователям снять стресс и улучшить их психологическое состояние.

### 4.3 Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

В таблице 1 приведена сравнительная характеристика с аналогами, наиболее близкими к теме разрабатываемого проекта.

	Antistress	Fidget Toys 3D	Pop It Fidget 3D	Satisfame	Fidget Toys 3d 2	Stress Ball	AntiStress	ZenUI
Наличие кнопки-переключателя	+	+	+	+	+	+	+	+
Наличие “Колыбели Ньютона”	+	-	-	+	+	+	+	+
Наличие «Concentration»	+	+	-	-	-	-	-	+

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 ТЗ 01-1				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

12  
**RU.17701729.05.01-01 ТЗ 01-1 01-1**

Наличие “Крякающей уточки”	+	-	-	+	-	-	-	+
Наличие спиннера	+	+	-	+	+	+	+	+
Наличие кликера	-	-	-	+	-	+	+	+
Наличие монетки	-	-	-	-	-	-	-	+
Наличие шара предсказаний	+	-	-	-	-	-	-	+
Наличие игральной кости	+	+	+	+	+	-	+	+
Наличие генератора случайных чисел	-	-	-	-	-	+	-	+
Наличие генератора игральной карты	-	-	-	-	-	-	-	+
Наличие ромашки	+	+	-	+	+	-	+	+
Наличие кнопки	+	+	-	+	+	+	+	+
Наличие слайдера	+	+	+	+	+	-	+	+
Наличие ксилофона	+	+	-	-	-	+	-	+
Наличие крестиков- ноликов	+	-	+	-	-	-	-	+
Наличие “Камня, ножниц, бумаги”	-	-	-	-	-	-	-	+
Наличие “2048”	-	-	-	-	-	-	-	+
Наличие “Четырех в ряд”	-	-	-	-	-	-	-	+
Наличие заметок	-	-	-	-	-	+	-	-
Возможность интеграции в	-	-	-	-	-	-	-	+

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 ТЗ 01-1				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

сторонние приложения								
-------------------------	--	--	--	--	--	--	--	--

Таблица 1 - Аналоги

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 ТЗ 01-1				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

- 1) ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 2) ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 3) ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 4) ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 5) ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 7) ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 8) ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 9) ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 10) ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 11) ГОСТ 15150-69 Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды. – М.: Изд-во стандартов, 1997.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 ТЗ 01-1				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

- 12) ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 13) Документация языка Swift. // URL: <https://developer.apple.com/documentation/swift> (дата обращения: 02.04.2024)
- 14) Федеральный закон “О персональных данных” от 27.07.2006 N 152-ФЗ. // URL: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_61801/](https://www.consultant.ru/document/cons_doc_LAW_61801/) (дата обращения: 02.04.2024)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 ТЗ 01-1				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и



**ПРИЛОЖЕНИЕ 1**

**ТЕРМИНОЛОГИЯ**

Таблица 1

Термин	Определение
2048	Игра, целью которой является получение плитки номинала «2048» (при желании можно продолжить дальше).
Concentration	Игра, в которой все карты выкладываются на поверхность лицевой стороной вниз, а две карты переворачиваются лицевой стороной вверх каждый ход.
MVVM	(Model-View-ViewModel) шаблон проектирования архитектуры приложения.
MVVM+C	(Model-View-ViewModel + Coordinator) шаблон проектирования архитектуры приложения. Отличием от MVVM является наличие Координатора, работа которого чаще всего сводится к созданию зависимостей.
Менеджер зависимости	Управляет зависимостями в разрезе конкретного проекта. То есть, в его задачи входит описание зависимостей, скачивание, обновление их исходного кода.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 ТЗ 01-1				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

**ПРИЛОЖЕНИЕ 2**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ**

Таблица 3

Пакет	Класс/Файл	Назначение
<b>ZenUILibrary</b>	FidgetToy	Класс для представления Fidget UI-элемента. Каждая игрушка имеет имя и изображение, которые используются для отображения.
	FidgetToyTitleView	Класс для отображения заголовка у UI-элемента
	RandomNumberGenerator	Один из классов, который отвечает за логику UI-элемента
<b>Screens/FidgetToyGridView</b>	FidgetToyGridView	Класс для отображения сетки на экране
	FidgetToyGridViewModel	Класс для модели представления сетки
<b>Screens/FidgetToyDetailView</b>	FidgetToyDetailView	Класс для отображения логики UI-элементов

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.01-01 ТЗ 01-1				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

[illegible]