



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03. (ПРИКЛАДНАЯ ИНФОРМАТИКА)

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОЙ РАБОТЕ

По дисциплине “Технология разработки программных систем”

НА ТЕМУ:

Android-приложение «Организатор будильников»

Студент ИУ6-54Б
(Группа)
Фамилия)

С.П.Пантелеев
(Подпись, дата) (И.О.

Руководитель курсовой работы

М.В.Широкова
(Подпись, дата) (И.О. Фамилия)

2020 г.

Реферат

РПЗ 58 страницы, 3 части, 28 рисунка, 11 таблиц, 5 источников, 3 приложения
БУДИЛЬНИК, ANDROID-ПРИЛОЖЕНИЕ, ОРГАНИЗАТОР, РАСПИСАНИЕ

Объектом разработки является Android-приложение “Организатор будильников”.

Цель работы – проектирование и реализация приложения “Организатор будильников”, позволяющий создавать будильники в соответствии с введенным расписанием, то есть времени первых пар, и заданного времени на путь в учебное заведение и времени сборы (встать, поесть, одеться и т.д.).

В результате разработки было спроектировано приложение позволяющее организовывать будильники, а также было проведено тестирование программного продукта методом структурного контроля, черного ящика и оценочного тестирования.

Разрабатываемое приложение создается как для студентов, так и для учащихся в любых других учебных заведений.

Содержание

Введение6

1. Анализ требований и уточнение спецификаций 7

1.1. Анализ задания и выбор технологии, языка и среды разработки 7

1.2. Разработка диаграммы вариантов использования 10

1.3. Разработка концептуальной модели предметной области 12

1.4 Разработка диаграмм деятельности 14

2. Проектирование структуры и компонентов программного продукта18

2.1. Разработка структурной схемы программного продукта18

2.2. Разработка интерфейса пользователя 18

2.2.1 Разработка форм интерфейса21

2.3. Разработка графов диалога26

2.4. Разработка алгоритмов создания будильников28

2.5. Разработка диаграммы компоновки38

3. Выбор стратегии тестирования и разработка тестов40

3.1 Тестирование структурным контролем40

3.2 Функциональное тестирование43

3.3 Оценочное тестирование55

Заключение57

Список используемых источников58

Приложение А. Техническое задание59

Приложение Б. Руководство пользователя.60

Приложение В. Листинг модуля TimeFirstLesson61

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ООП – объектно-ориентированное-программирование.

ТЗ – техническое задание.

ЧС – числитель, нечетная неделя.

ЗН – знаменатель, четная неделя.

Android – мобильная операционная система.

АРК файл – итоговый исполняемый файл в системе android.

Интерфейс – графическое представление приложения пользователю.

ПП – программный продукт.

Введение

Работа посвящена проектированию и разработке Android-приложения “организатор будильников” для мобильных платформ под управлением операционной системы Android. Разрабатываемое приложение может быть использовано студентам вузов, так и другими людьми, обучающиеся в других образовательных местах. Программный продукт позволит создавать будильники для учебного расписания в зависимости введенных дополнительных настроек.

В ходе выполнения работы были рассмотрены аналоги, которые предоставляют стандартные действия по созданию будильников, однако имеющие различные интерфейсы и дополнительные настройки.

Актуальность разработки приложения заключается в том, что в рассмотренных аналогах нигде не было найдено такого, которое могло бы быть удобно для учащихся и подстраивалось под их учебное расписание. Разрабатываемое приложение, предоставляет пользователю возможность задать, времена начала первых пар учебного расписания и дополнительные настройки, которые включают в себя: время на встать (поесть, одеться и так далее), времена на путь в учебное заведение, необходимое количество будильников на встать и на выход. Это позволит пользователю организовывать будильники на учебную неделю с расчетом всех требований по временным рамкам.

1. Анализ требований и уточнение спецификаций

1.1. Анализ задания и выбор технологии, языка и среды разработки

Для уточнения требований, указанных в ТЗ, был проведен анализ предметной области, который заключался в рассмотрении существующих аналогов систем создания будильников. Одними из самых популярных являются:

- “Простой будильник бесплатно” от Moula Software;
- “Будильник” от RV AppStudios;
- “Будильник” от Javier Salmona;
- “Будильник Wakey” от cookiebits.io;
- “Умный будильник” от Smart Mobile Tools;
- “Будильник Early Bird” от 1year;
- “Будильник для любителей поспать - AMdroid” от AMdroid Alarm Clock - No more oversleep.

Небольшое описание каждого из них показано в таблице 1. Сравнительный же анализ приложений показан в таблице 2.

Таблица 1 – Описание аналогов систем формирования будильников

Название	Описание
“Простой будильник бесплатно” от Moula Software.	Предоставляет стандартное приложение по созданию будильника.
“Будильник” от RV AppStudios.	Помимо обычного будильника может предложить на выбор то, как пользователь будет выключать будильник.
“Будильник” от Javier Salmona.	Предлагает картинки зверьков, которые будут нарисованы рядом со списком будильников.
“Будильник Wakey” от cookiebits.io.	При создании будильника предлагает посмотреть время восхода и заката.
“Умный будильник” от Smart Mobile Tools	Включает в себя небольшой блокнот для записей и предлагает математическую задачу для отключения будильника.
“Будильник Early Bird” от 1year.	Предоставляет стандартное приложение по созданию будильника.

Продолжение таблицы 1

Название	Описание
“Будильник для любителей поспать - AMdroid” от AMdroid Alarm Clock - No more oversleep.	Можно поставить время в виде ночника и функция срабатывания будильников только в указанных реальных районах земного шара

Таблица 2 – Сравнительный анализ аналогов

Критерий \ Приложение	Создание будильников	Различны-е способы выключения будильников	Смена ди-зай на	Учет учебного расписа-ния учащихся	Автоматическая организация п-ого количества будильников	Отсутствие платного контента или рекламы
Разрабатываемое приложение «Организатор будильников»	+	-	-	+	+	+
“Простой будильник бесплатно” от Moula Software.	+	-	-	-	-	-
“Будильник” от RV AppStudios.	+	+	+	-	-	-
“Будильник” от Javier Salmona.	+	-	+	-	-	-
“Будильник Wakey” от cookiebits.io.	+	-	-	-	-	-
“Умный будильник” от Smart Mobile Tools	+	+	-	-	-	-

Продолжение таблицы 2

Критерий \ Приложение	Создание будильников	Различные способы выключения будильников	Смена ди-зайна	Учет учебного расписания учащихся	Автоматическая организация п-ого количества будильников	Отсутствие платного контента или рекламы
“Будильник Early Bird” от 1year.	+	-	-	-	-	-
“Будильник для любителей поспать - AMdroid” от AMdroid Alarm Clock - No more oversleep.	+	-	-	-	-	-

При рассмотрении аналогов можно сделать вывод, что приложения почти не отличаются друг от друга. Однако, некоторые аналоги имеют уникальные опции такие как: просмотр заката и рассвета, настройка места срабатывания будильника, встроенный блокнот или различные способы выключения будильников. Также каждый из аналогов имеет рекламу или платные опции. Почти все перечисленные функции будут перегружать приложение в целом, и интерфейс в частности, что может вызвать трудности у пользователя при использовании приложения и дополнительную нагрузку на систему телефона.

В результате изучения аналогов не было найдено приложения для студентов или других учащихся, которое бы создавало будильники в соответствии с заданным временем первых пар учебного расписания и дополнительными параметрами, такими как:

- время на путь в учебное заведение;

- время на сборы, то есть время, которое необходимо пользователю чтобы встать, одеться, поесть и так далее;
- количество будильников на встать;
- количество будильников на сборы;
- временной промежуток между будильниками на встать;
- временной промежуток между будильниками на сборы.

В качестве технологии программирования было выбрано объектно-ориентированное программирование. ООП является основой всех современных приложений и имеет удобное и практическое применение. При использовании этого метода вся программа разбивается на объекты, с каждым из которых работают по отдельности.

Для разработки приложения на операционной системе андроид самыми распространёнными языками являются Java, Kotlin и C#. Один из самых развитых, и имеющий больше пособий и нотаций от разработчиков в сфере android сейчас – Java, который и был выбран в качестве языка программирования, также он во многом схож с C++ как по синтаксису, так и по ООП подходу [1].

Средой разработки была выбрана одна из самых распространённых IDE для разработки под android - Android Studio. IDE предоставляет удобный инструмент для разработки ПО для android, в виде графического окна настройки внешнего вида интерфейса приложения, а также функцию эмулятора телефона с возможностью выбора разных версий android и телефонов (более слабых и более мощных).

Приложение имеет MVC - Model-View-Controller, то есть модель-представление-контроллер архитектуру, где view — это activity, являющимся визуальным интерфейсом приложения, а model – это классы, где происходит вся работа приложения. Через controller происходит обмен информацией между activity и классами, то есть controller говорит классу о том, что в activity была нажата какая-то кнопка, на что класс среагирует так, как задано программным кодом. После выполнения кода, если что-то надо отобразить в интерфейсе, эта информация передается сразу из model в view и показывается пользователю [2].

1.2. Разработка диаграммы вариантов использования

Первым шагом при проектировании приложения является определение основных вариантов ее использования, для этого была разработана диаграмма вариантов

использования, которая отражает варианты взаимодействия пользователя с разрабатываемым приложением [3].

После анализа технического задания были получены следующие варианты взаимодействия пользователя и приложения:

- ввод времени первых пар;
- ввод дополнительных настроек;
- выбор вида расписания;
- создание будильников;
- смена недели;
- просмотр недели.

Рассмотрим основной вариант использования, который описан в таблицах 3, 4.

Таблица 3 – описание вариантов использования *Создание будильников*

Название варианта	Создание будильников
Цель	Создание новых будильников
Действующие лица	Пользователь
Краткое описание	Ввод необходимых данных и на их основе создание будильников
Тип	Основной

Таблица 4 – Вариант использования *Создание будильников*

Действие пользователя	Отклик системы
1. Пользователь вводит дополнительные настройки и нажимает кнопку “запомнить”.	2. Приложение запоминает введенные дополнительные настройки.
3. Пользователь выбирает: а) обычное расписание; б) расписание МГТУ.	4. Приложение открывает окно: а) с обычным расписанием, см. раздел Ввод данных в обычном расписании (таблица 5); б) с расписанием МГТУ, см. раздел Ввод данных в расписании МГТУ (таблица 6).
5. Пользователь вводит необходимые данные и нажимает кнопку “создать будильники”.	6. Приложение создает будильники со всеми введенными пользователем данными.

Таблица 5 – раздел *Ввод данных в обычном расписании*

Действие пользователя	Отклик системы
1. Пользователь вводит время первых пар в необходимые ему дни.	

Таблица 6 – раздел **Ввод данных в расписании МГТУ**

Действие пользователя	Отклик системы
1. Пользователь выбирает ЧС\ЗН и вводит время первых пар в необходимые для пользователя дни и нажимает кнопку запомнить	2. Приложение запоминает введенное время на указанную неделю.
3. Пользователь выбирает другую неделю и снова вводит время первых пар в необходимые для пользователя дни и нажимает кнопку запомнить.	4. Приложение запоминает введенное время на указанную неделю.
5. Пользователь выбирает текущую неделю	

Разработанная диаграмма вариантов использования представлена на рисунке 1.

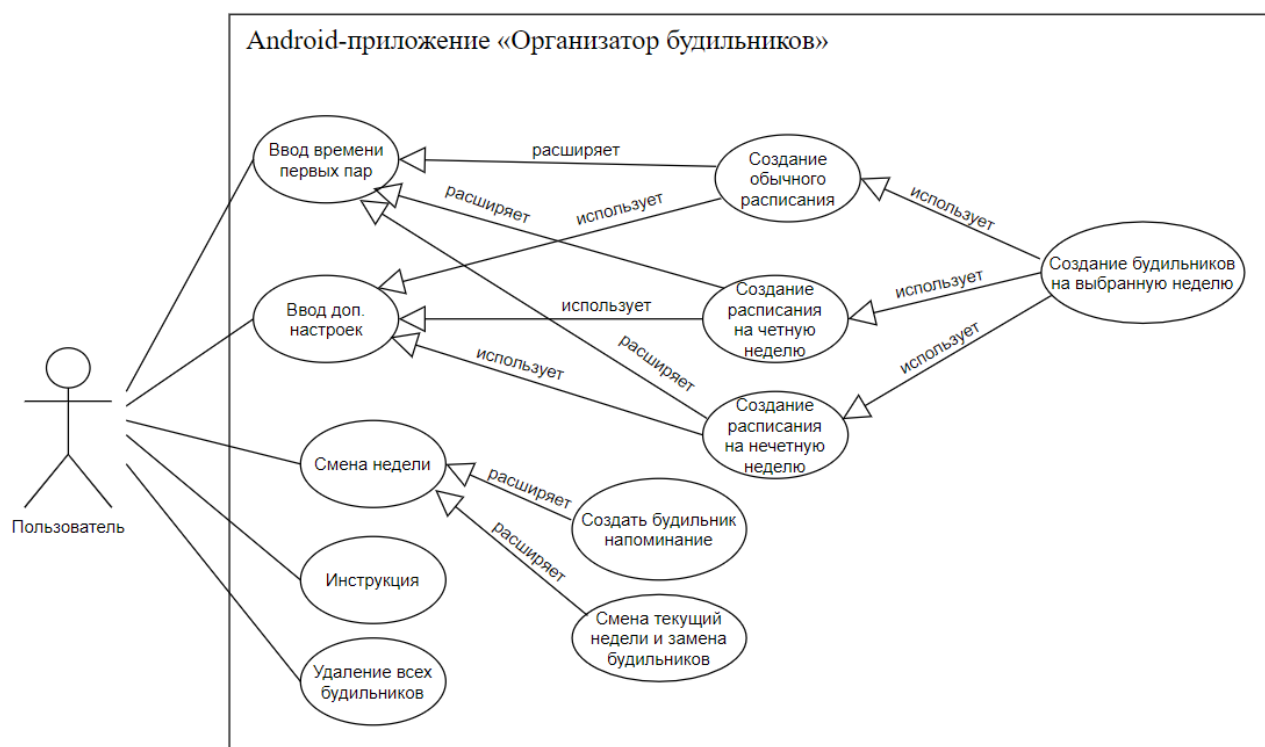


Рисунок 1 – Диаграмма вариантов использования

1.3. Разработка концептуальной модели предметной области

При анализе предметной области и технического задания были выявлены следующие основные объекты:

- список будильников;
- данные;
- вид расписания:
 - обычное;
 - четная и нечетная неделя;
- время первых пар;
- дополнительные настройки:
 - время на сборы;
 - время на путь;
 - количество будильников;
 - промежутки между будильниками;
- меню;
- инструкция.

На их основе была построена концептуальная модель предметной области, которая продемонстрирована на рисунке 2 [3].

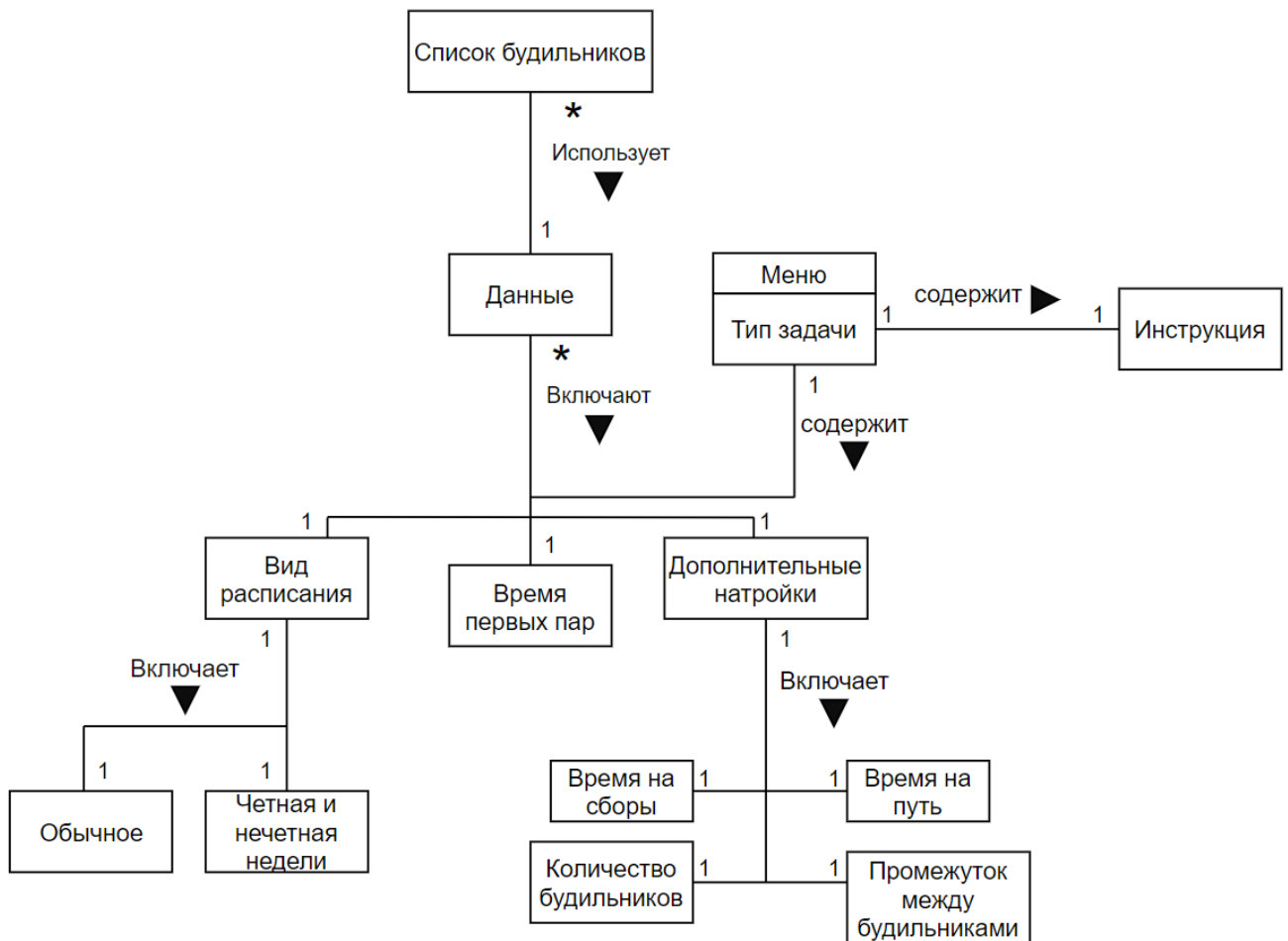


Рисунок 2 - Концептуальная модель предметной области

Меню содержит несколько типов задач - дополнительные настройки, время первых пар, расписание (обычное и МГТУ) и инструкцию.

Дополнительные настройки включают в себя время на сборы (встать, поесть, одеться и так далее), время на путь, количество будильников и промежуток между этими будильниками. Все эти параметры будут учитываться при создании будильников.

Вид расписания содержит обычное и МГТУ расписание. В обычном предлагается неделя со статичным учебным планом, а в расписании МГТУ ввод данных на четную и нечетную неделю.

Создание будильников использует данные из приложения, для создания тех будильников, которые необходимы именно пользователю. Эти данные формируются из того, какое расписание выбрано, времени первых пар, и дополнительных настроек.

1.4 Разработка диаграмм деятельности

Для подробного описания взаимодействий пользователя и приложения, были разработаны диаграммы деятельности для некоторых функций, а именно:

- создание будильников в обычном расписании;
- создание будильников в расписании МГТУ.

Диаграмма деятельности для создания будильников в обычном расписании изображена на рисунке 3. Диаграмма показывает, что пользователь для начала работы с приложением должен ввести дополнительные параметры (время на сборы и на выход, количество будильников и промежуток между ними), если первые два параметра введены, то производится переход на страницу обычного расписания. После чего пользователь вводит время первых пар в необходимые ему дни и нажимает кнопку “создать будильники”, на чем этот процесс заканчивается успешным созданием будильников.

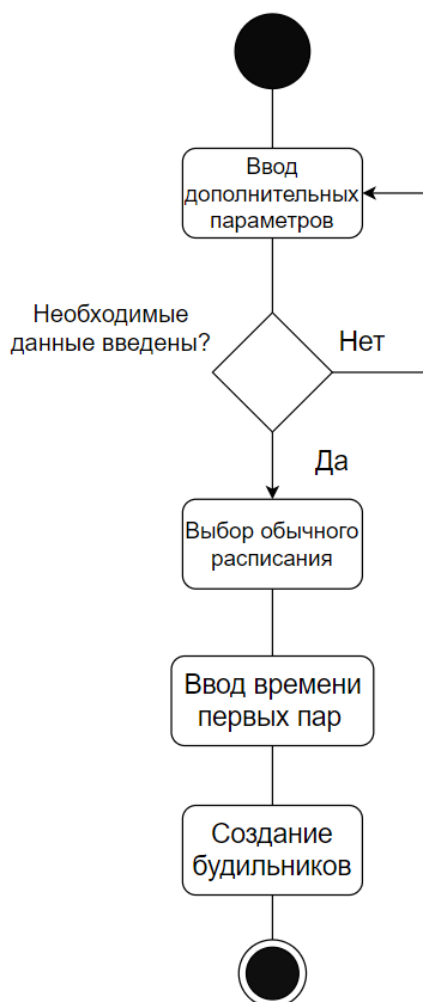


Рисунок 3 - Диаграмма деятельности для создания будильников в обычном расписании

Если пользователь сначала ввел время первых пар, и нажал кнопку “создать будильники”, то приложение выведет уведомление, что не введены дополнительные настройки. После чего пользователь должен ввести дополнительные параметры (время на сборы и на выход, количество будильников и промежутков между ними), если первые два параметра введены, то пользователь снова должен ввести время первых во вкладке обычного расписания и нажать кнопку “создать будильники”.

Диаграмма деятельности по созданию будильников для расписания МГТУ показана на рисунке 4. Она показывает, что для начала работы с приложением пользователь должен ввести дополнительные параметры (время на сборы и на выход, количество будильников и промежутков между ними), если первые два параметра введены, то производится переход на страницу расписания МГТУ. Далее пользователь выбирает одну из недель (ЧС\ЗН), на которую будет задавать время первых пар, вводит их, запоминает, с помощью одноименной кнопки, выбирает другую неделю и повторяет действие. После чего выбирает текущую неделю и создает будильники. На этом процесс заканчивается.

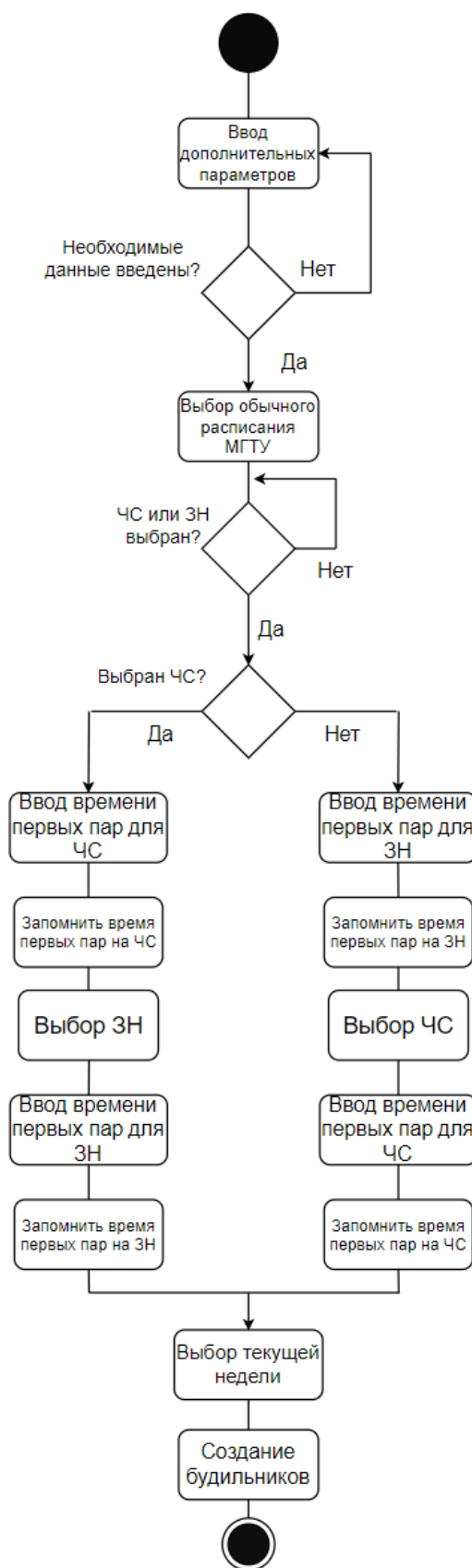


Рисунок 4 - Диаграмма деятельности по созданию будильников для расписания МГТУ

Пользователь может сначала выбрать одну из недель (ЧС\ЗН), на которую будет задавать время первых пар, вводит их, и запоминает введенное время с помощью

одноименной кнопки, так же он может выбрать другую неделю и повторить действия, после чего выбрать текущую неделю и создает будильники, но приложение выведет уведомление, что не введены дополнительные настройки. Тогда пользователь должен ввести дополнительные параметры (время на сборы и на выход, количество будильников и промежуток между ними), если первые два параметра введены, то пользователь возвращается в окно расписание МГТУ, выбирает текущую неделю и нажимает кнопку “создать будильники”.

2. Проектирование структуры и компонентов программного продукта

2.1. Разработка структурной схемы программного продукта

Структурная схема программного продукта показывает деление программы на её главные составляющие. На основе анализа технического задания в разрабатываемом приложении, которое организует время будильников для учащихся, выявлено четыре основных подсистемы:

- в подсистеме организации будильников обычного расписания создаются будильники в соответствии с расписанием обычного студента, имеющий фиксированный тип недели;
- в подсистеме дополнительных настроек времени происходит ввод дополнительных параметров для дальнейшего создания будильников;
- в подсистеме организации будильников расписания МГТУ создаются будильники в соответствии с расписанием студента МГТУ, то есть с специфичной системой смены недель – числителя и знаменателя;
- в подсистеме смены недели происходит замена текущих будильников, на будильники альтернативной недели. Работает только для будильников МГТУ.

На основе выявленных подсистем была составлена структурная схема ПП, показанная на рисунке 5.

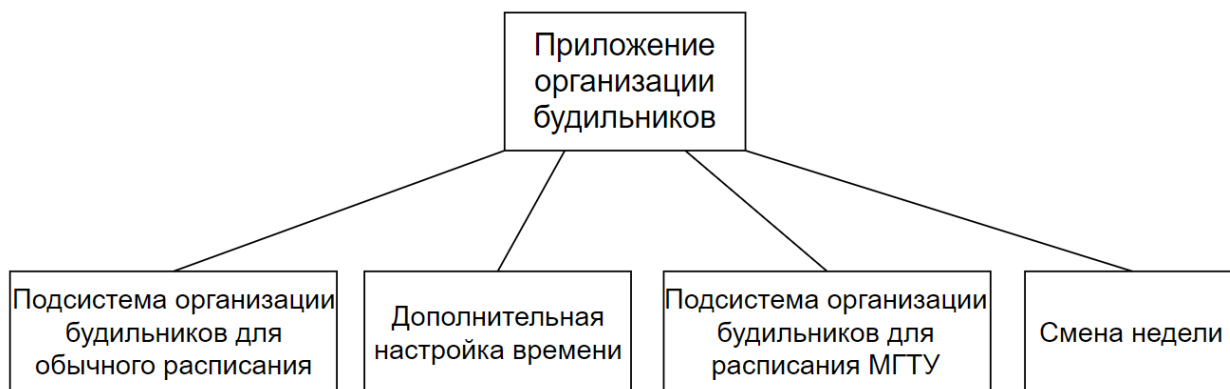


Рисунок 5 – Структурная схема программного продукта

2.2. Разработка интерфейса пользователя

В следствии анализа предметной области был выбран вид интерфейса со свободной навигацией, то есть пользователь может свободно перемещаться по окнам приложения и

смотреть, что программный продукт может предложить. Так устроены почти все современные приложения, и пользователи привыкли, что могут перемещаться по разным вкладкам интерфейса. Разработанная диаграмма состояний интерфейса, для наглядного отображения, представлена на рисунке 6 [3].

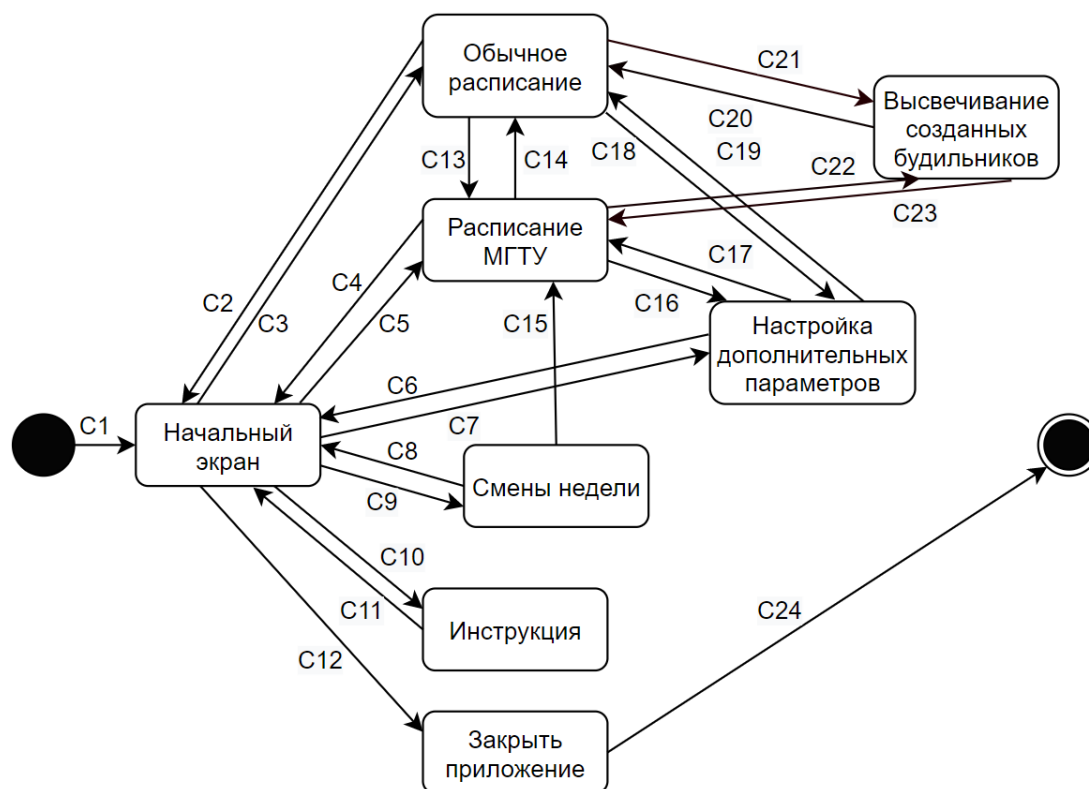


Рисунок 6 – Диаграмма состояний интерфейса

C1 – активация формы при запуске приложения;

C2 – возвращение на главный экран по нажатию кнопки “вернуться на начальный экран”;

C3 – по нажатию кнопки “Расписание” переход в окно стандартного расписание;

C4 – возвращение на главный экран по нажатию кнопки “вернуться на начальный экран”;

C5 – по нажатию кнопки “Расписание МГТУ” переход в окно расписания МГТУ;

C6 – возвращение на главный экран по нажатию кнопки “вернуться на начальный экран”;

C7 – по нажатию кнопки “Настройка дополнительных параметров” переход в окно настройки дополнительных параметров времени;

C8 – возвращение на главный экран по нажатию кнопки “вернуться на начальный экран”;

- C9 – по нажатию кнопки “Смена недели” переход в окно смены текущей недели;
- C10 – возвращение на главный экран по нажатию кнопки “вернуться на начальный экран”;
- C11– возвращение на главный экран по нажатию кнопки “вернуться на начальный экран”;
- C12 – по нажатию кнопки “Заккрыть приложение” приложение прекращает свой текущий сеанс работы;
- C13 – переход в расписание МГТУ из стандартного по нажатию кнопки “Иное расписание”;
- C14 – переход в стандартного расписание из расписания МГТУ по нажатию кнопки “Обычное расписание”;
- C15 – переход в расписание МГТУ из окна смены недели по нажатию кнопки “Расписание МГТУ”;
- C16 – переход в дополнительные настройки из стандартного расписания по нажатию кнопки “Настройка доп. времени”;
- C17 – переход в обычное расписание из дополнительных настроек по нажатию кнопки “Расписание”;
- C18 – переход в дополнительные настройки из расписания МГТУ по нажатию кнопки “Настройка доп. времени”;
- C19 – переход в расписание МГТУ из дополнительных настроек по нажатию кнопки “Расписание МГТУ”;
- C20 – переход в расписание МГТУ из окна с созданными будильниками по нажатию кнопки “Назад”;
- C21 – переход в окно с созданными будильниками из обычного расписания по нажатию кнопки “Создать будильники”;
- C22 – переход в окно с созданными будильниками из расписания МГТУ по нажатию кнопки “Создать будильники”;
- C23 – переход в обычное расписание из окна с созданными будильниками по нажатию кнопки “Назад”;
- C24 – закрытие приложения.

Учащийся пользователь может быть не только студентом из МГТУ, поэтому на главной странице предлагается выбор между обычным, то есть статичном расписанием,

которое не меняется, и расписанием МГТУ, где специфическая система разделения расписания по четным и нечетным неделям.

При вводе времени первых пар и создании будильника пользователь может забыть ввести дополнительные настройки, и для этого есть возможность перехода к ним из окна обоих расписаний, минуя переход на главный экран.

Функция смены недели относится к расписанию МГТУ, поэтому из окна смены недели можно перейти в расписание МГТУ, минуя переход на главный экран, чтобы поменять установленное время, если вдруг расписание на следующую неделю поменялось.

2.2.1 Разработка форм интерфейса

Приложение имеет прямое взаимодействие с пользователем, поэтому надо сделать удобный и интуитивно понятный интерфейс. Разработанные формы интерфейса проекта показан на рисунках 7 – 12 [4].

Приложение имеет некоторое количество опций, чтобы пользователю было удобно выбирать, что именно ему нужно, при заходе в приложение, было создано главное меню, показанное на рисунке 7.

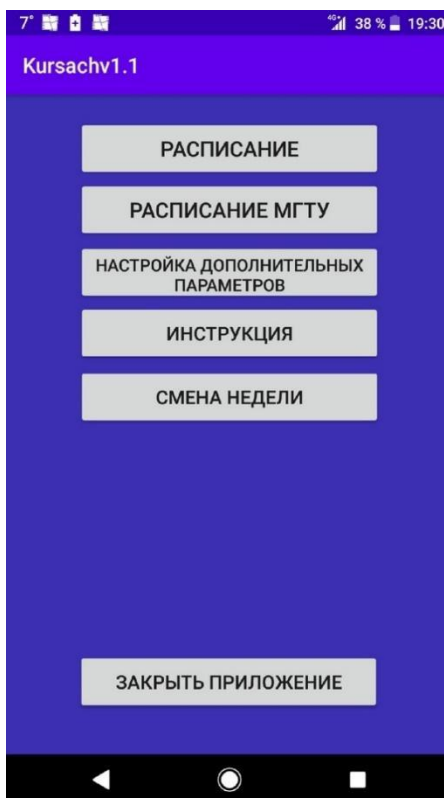


Рисунок 7 – Окно главного меню

Окно “обычное расписание”, которое показано на рисунке 8, содержит поля ввода времени первых пар и кнопки:

- “Иное расписание”, чтобы перейти на расписание МГТУ, если вдруг пользователь ошибся расписанием;
- “Создать будильники”, чтобы после ввода времени сразу создать будильники;
- “Удалить будильники”, чтобы можно было удалить будильники и сразу создать новые;
- “Настройка доп. параметров”, чтобы можно было в них перейти и ввести необходимые данные, минуя переход в главное меню, если что-то ранее было не введено;
- “Вернуться на начальный экран”, что пользователь мог вернуться в главное меню увидеть все опции приложения.



Рисунок 8 – Окно обычного расписания

Окно “обычное расписание”, которое показано на рисунке 9, содержит поля ввода времени первых пар и кнопки:

- “Иное расписание”, чтобы перейти в обычное расписание, если вдруг пользователь ошибся расписанием;
- “Создать будильники”, чтобы после ввода времени сразу создать будильники;
- “Удалить будильники”, чтобы можно было удалить будильники и сразу создать новые;
- “Настройка доп. параметров”, чтобы можно было в них перейти и ввести необходимые данные, минуя переход в главное меню, если что-то ранее было не введено;
- “Вернуться на начальный экран”, что пользователь мог вернуться в главное меню и снова увидеть все опции приложения.

И два переключателя, чтобы выбирать какая сейчас неделя, одновременно видя, какое время пользователь ввел.

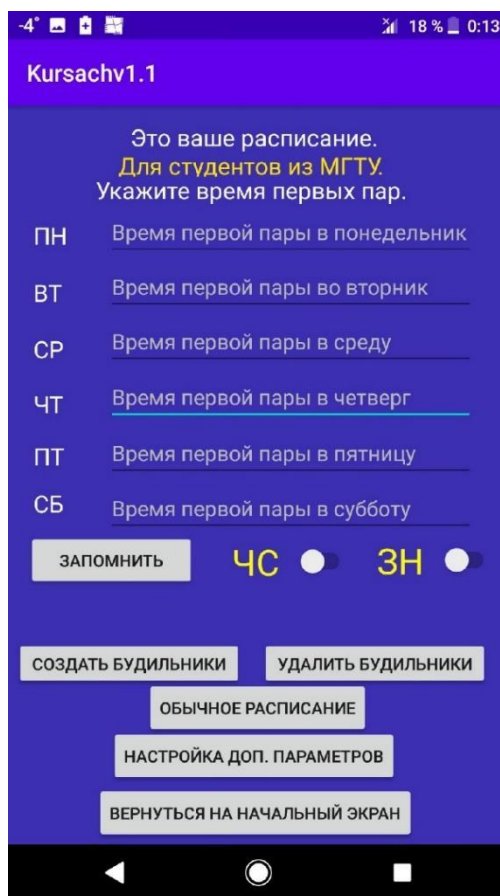


Рисунок 9 – Окно расписания МГТУ

Окно “Настройка дополнительных параметров”, которое показано на рисунке 10, содержит поля ввода:

- времени необходимые пользователю на сборы и на выход;
- количества будильников на встать и на выход;

- промежуток времени между будильниками на встать и на выход;
- и кнопки:
- “Расписание”, чтобы перейти в обычное расписание, если вдруг пользователь ошибся расписанием;
- “Расписание МГТУ”, чтобы перейти расписание МГТУ, если вдруг пользователь ошибся расписанием;
- “ Вернуться на начальный экран”, что пользователь мог вернуться в главное меню и снова увидеть все опции приложения.

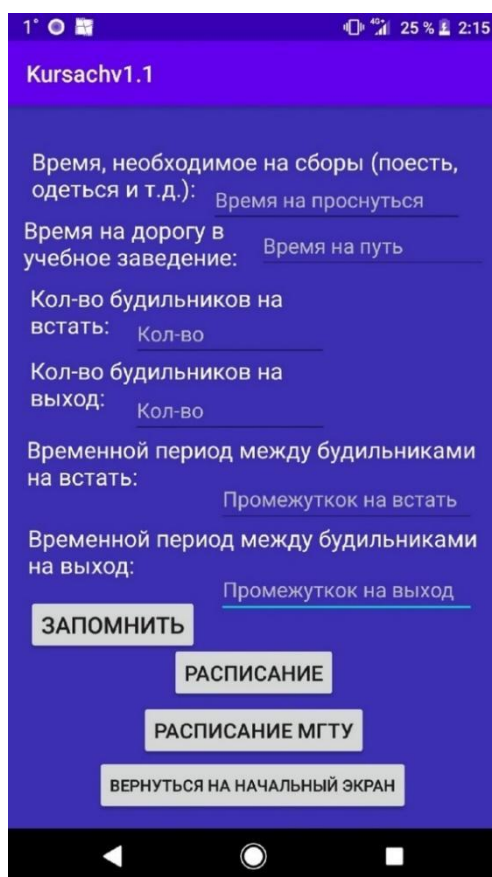


Рисунок 10 – Окно настройки дополнительных параметров

Окно “Смена недели”, которое показано на рисунке 11, содержит кнопку “Смену недели”, для смены будильников с ЧС на ЗН или наоборот, поле ввода времени будильника и соответствующую кнопку “Создать будильник”, чтобы можно было создать будильник, который будет напоминать о смене недели и кнопку “Вернуться на начальный экран”, которая позволит пользователю вернуться в главное меню и снова увидеть все опции приложения.

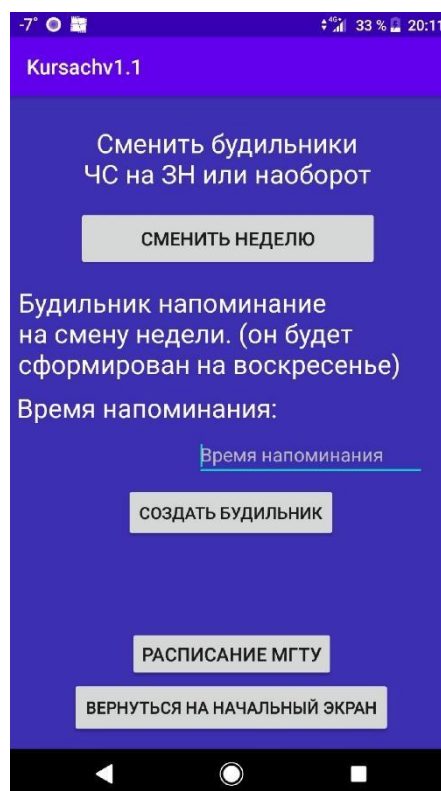


Рисунок 11 – Окно смена недели

Окно “Инструкция”, которое показано на рисунке 12, показывает краткую инструкцию о корректном использовании приложения. И кнопку “Вернуться на начальный экран”, которая позволит пользователю вернуться в главное меню и снова увидеть все опции приложения.

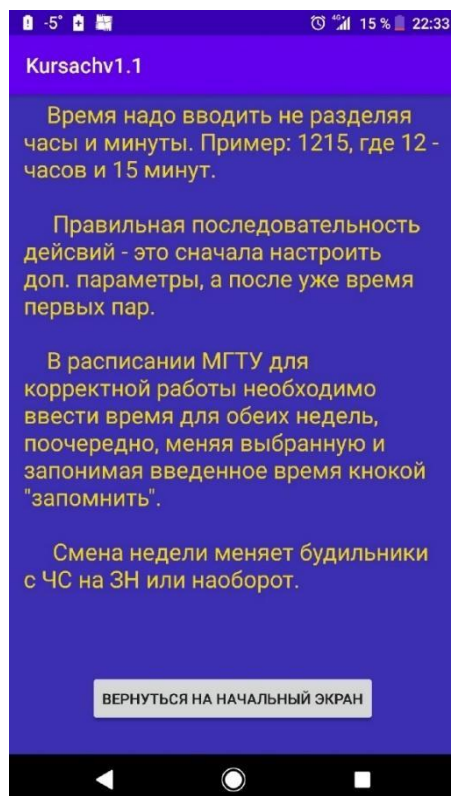


Рисунок 12 – Окно инструкция

Во время использования приложения пользователь может предпринимать разные действия, которые могут привести к тому, что не заложено в работе приложения, интерфейс должен показывать, что допущена ошибка и причину этой ошибки. В разрабатываемом приложении выдача ошибок пользователю реализована с помощью функции сплывающих сообщений. Примеры такого уведомления показан на рисунке 13.

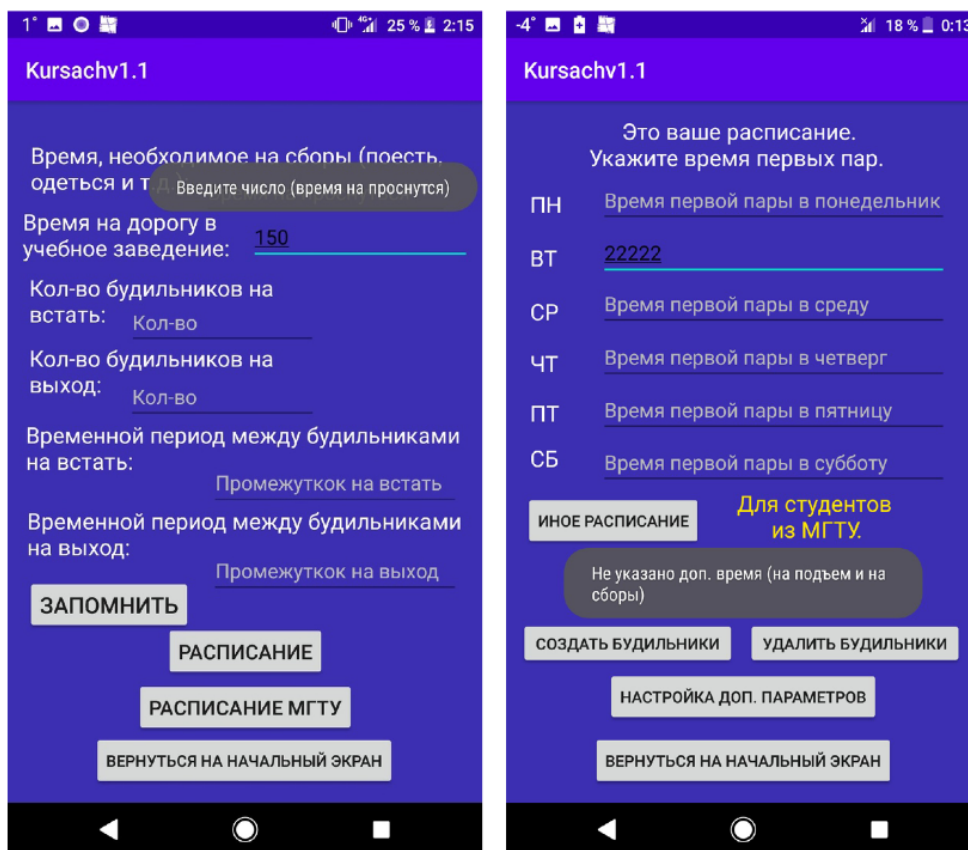


Рисунок 13 – Уведомления о некорректном вводе

2.3 Разработка графов диалогов

Разрабатываемое приложение имеет несколько окон, следовательно диалог приложения и пользователя не всегда будет одинаковым. Для удобного взаимодействия пользователя и приложения необходимо продумать как приложение будет реагировать на различные действия пользователя, чтобы это более подробно показать были разработаны несколько графов диалогов:

- общий граф диалога;
- граф для настройки дополнительных параметров;
- граф для создания будильников для обычного расписания;

- граф для создания будильников для расписания МГТУ.

Общий граф диалога показан на рисунке 14, который показывает работу приложения в целом. То есть сначала пользователь выбирает, куда хочет перейти. После производит ввод времени пар, дополнительных настроек и выбора алгоритма (обычное расписание или МГТУ). Далее уже создаются будильники, и пользователь видит уведомление об этом.

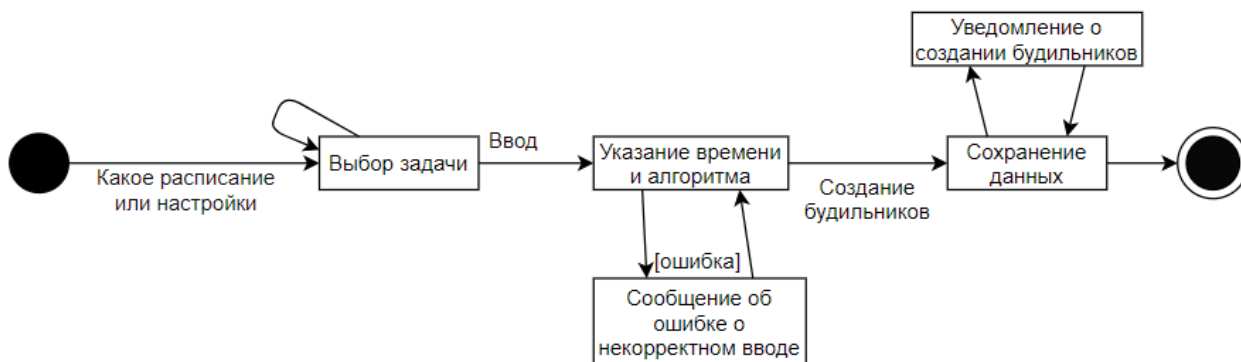


Рисунок 14 – Общий граф диалога

Граф для настройки дополнительных параметров показан на рисунке 15, который показывает диалог с пользователем во время ввода дополнительных настроек будильников. Пользователь вводит время на путь в универ, время на сборы, количество необходимых будильников и интервалы между ними, если какие-то данные введены некорректно, то система ему об этом сообщает. После данные сохраняются в системе и передаются в поля классов, которые именно создают будильники и всплывает уведомление, что правильно введенные данные запомнены.

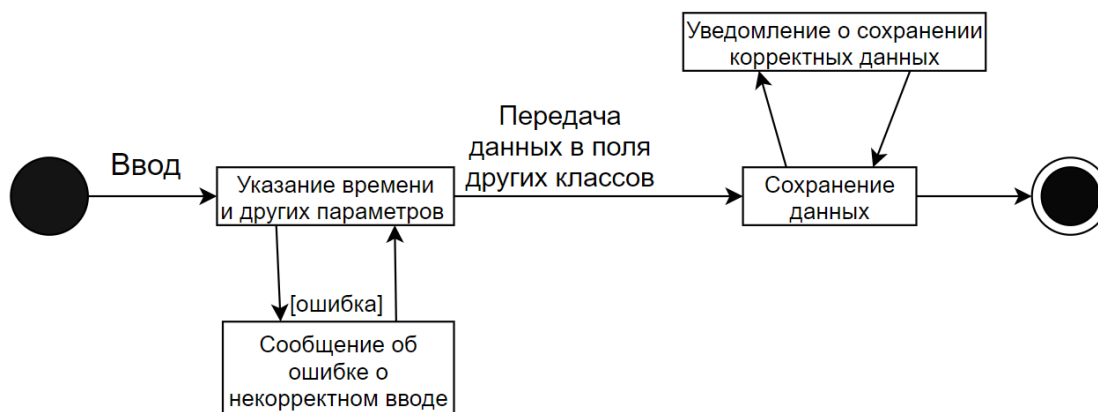


Рисунок 15 – Граф для настройки дополнительных параметров

Граф для создания будильников для обычного расписания показан на рисунке 16, который показывает работу диалог с пользователем во время создания будильников в

обычном расписании. Пользователь вводит время первых пар, если какие-то данные введены не корректно, то система ему об этом сообщает. Если же не введены обязательные данные дополнительных настроек, то система тоже ему об это говорит. Если же все данные есть, и они правильные, то создаются будильники и высвечивается уведомление об этом.

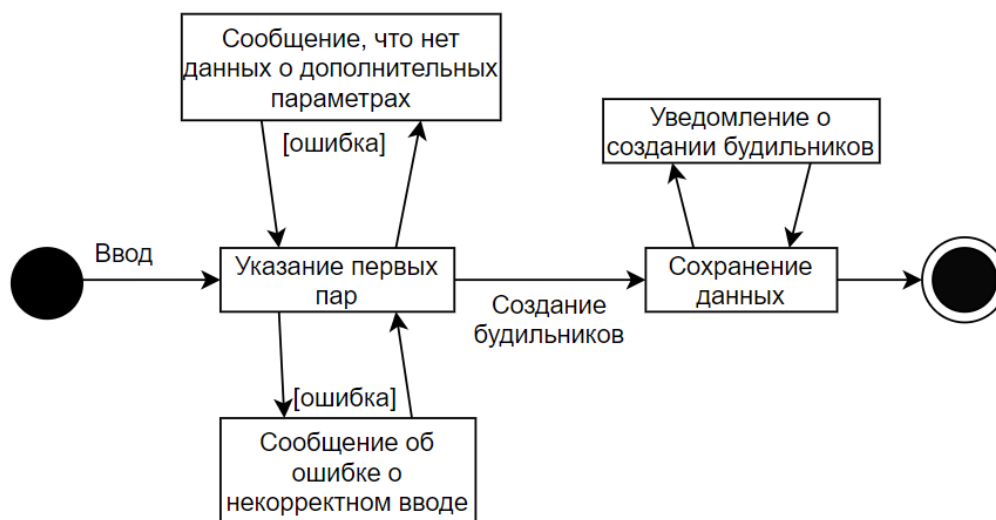


Рисунок 16 – Граф для создания будильников для обычного расписания

Граф для создания будильников для расписания МГТУ показан на рисунке 17, который показывает диалог при создании будильников для расписания студента МГТУ. Пользователь выбирает одну из недель (числитель\знаменатель), вводит время первых пар на эту неделю, если какие-то данные введены не корректно, то система ему об этом сообщает, или если же не введены обязательные данные дополнительных настроек, то система тоже ему об это говорит. После пользователь меняет выбранную неделю и повторяет свои действия, если какие-то данные введены не корректно, то система ему об этом сообщает. Далее пользователь выбирает текущую неделю и на выбранную неделю создаются будильники.

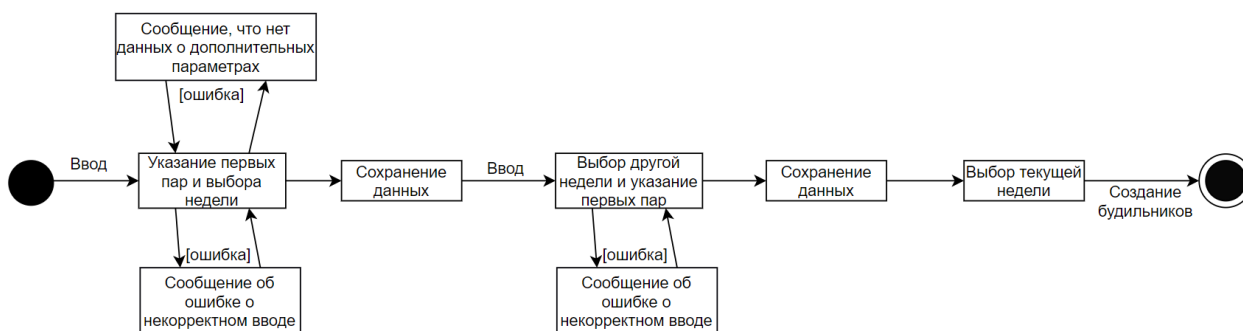


Рисунок 17 – Граф для создания будильников для расписания МГТУ

2.4. Разработка алгоритмов создания будильников

Будильники в разрабатываемом проекте являются основополагающей частью приложения, и для разработки важно понимать, как будет работать этот механизм изнутри, чтобы все было корректно и точно. Так как приложение рассчитано как для обычных студентов, так и для студентов МГТУ, то нужно проработать оба алгоритма по созданию будильников. Общий алгоритм работы в целом и алгоритм настройки дополнительных параметров в частности, у обоих расписаний одинаковый. Но отличается алгоритм ввода и запоминание первых пар и соответственно будет отличаться почти весь алгоритм формирования будильников. Общий алгоритм работы и алгоритм настройки дополнительных параметров показан на рисунках 18 и 19 соответственно.

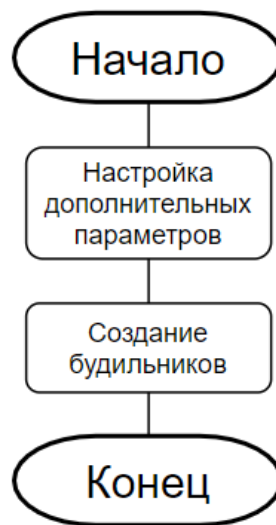


Рисунок 18 – Общий алгоритм работы

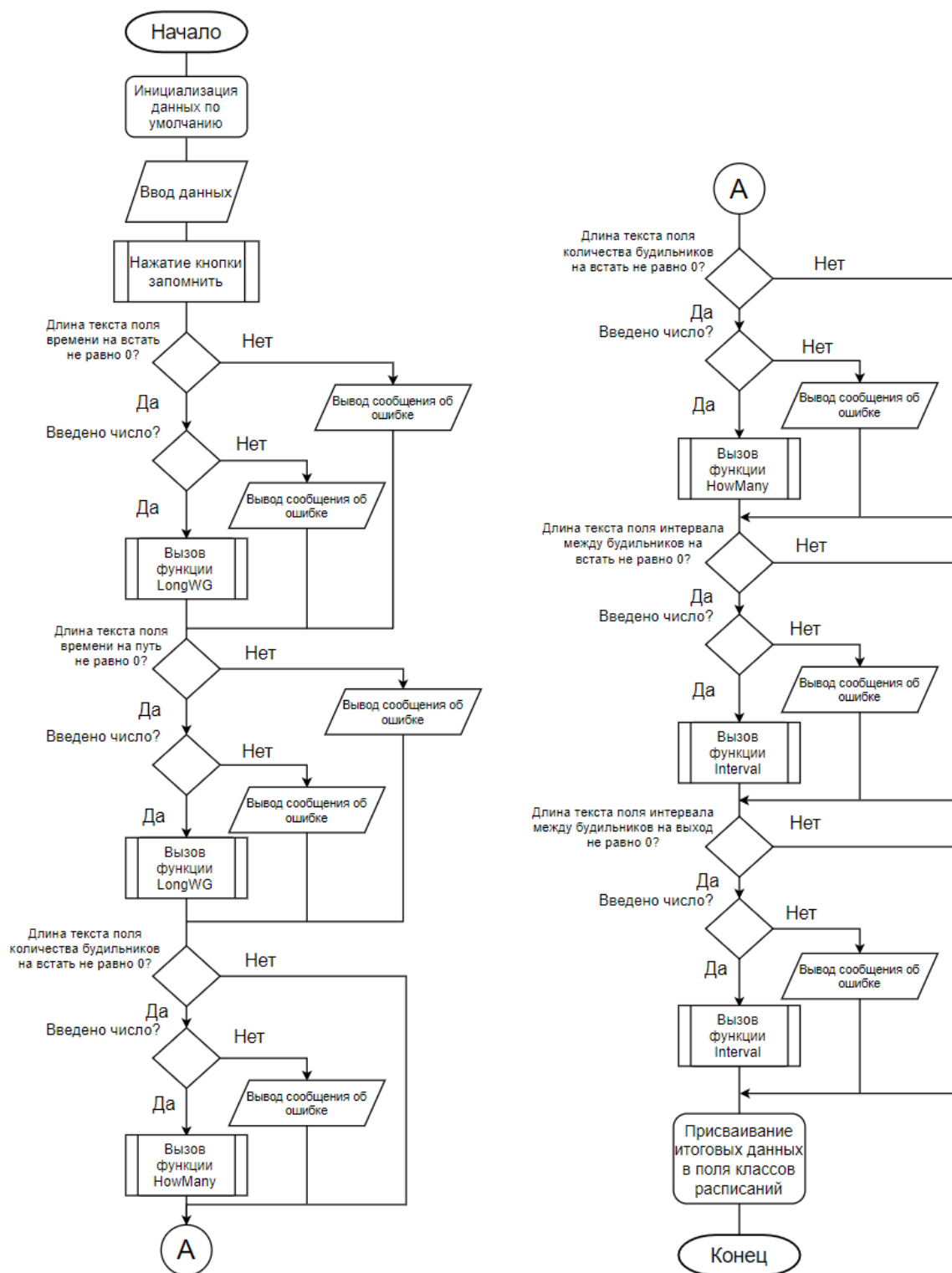


Рисунок 19 – Алгоритм настройки дополнительных параметров

Подпрограмма LongWG отвечает за проверку введенных данных в поля “Время, необходимое на сборы” и “Время на дорогу в учебное заведение”. Сначала производится проверка, что введено положительное число, далее, что число минут меньше 60, так как такого формата времени не существует. После идет проверка что оба промежутка не были равны по 4 часа, так как в этом случае первый будильник будет за 8 часов до пары, и если

начало пары в 8 часов, то на время 00:00 приложение не сможет определить на какой день ставить будильник. Алгоритм работы подпрограммы LongWG показан на рисунке 20.

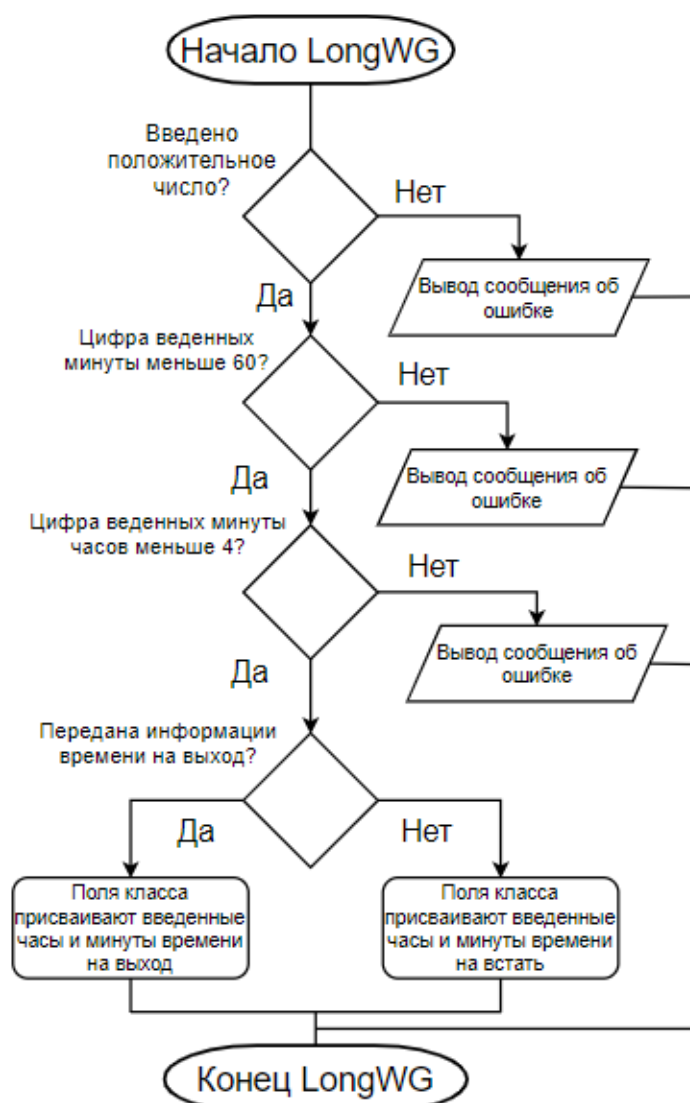


Рисунок 20 – Алгоритм работы подпрограммы LongWG

Подпрограмма NowMany отвечает за проверку введенных данных в поля “Кол-во будильников на встать” и “Кол-ва будильников на выход”. Чтобы число будильников было больше 0, иначе ничего не создаться, и меньше 10, так как при случайном большом вводе система может не создать введенное количество будильников, и будет занято место для нужных будильников, которые не создадутся, так как все пространство памяти под будильники уже будет занято. Алгоритм работы подпрограммы показан на рисунке 21.

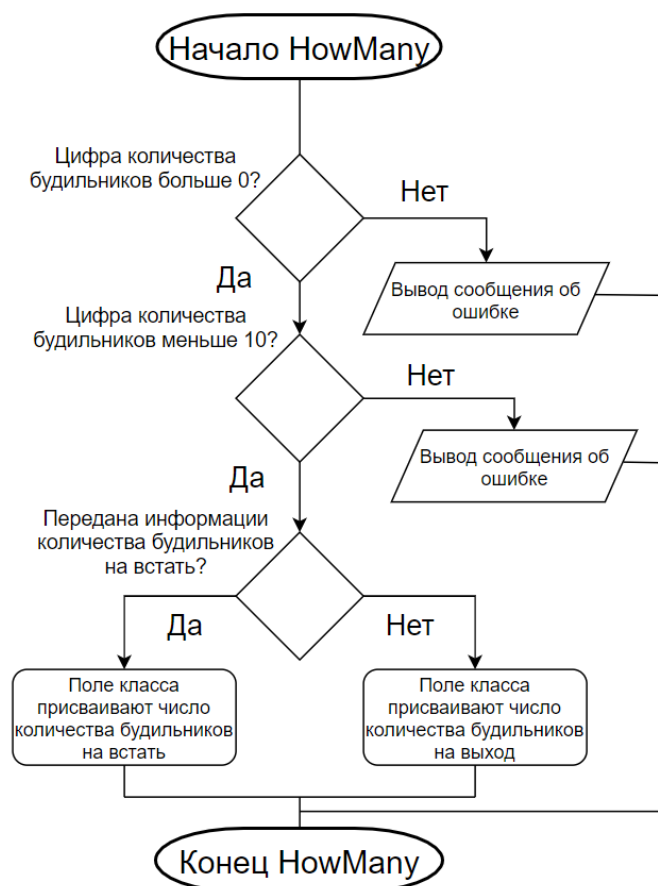


Рисунок 21 – Алгоритм работы подпрограммы HowMany

Подпрограмма Interval отвечает за проверку введенных данных в поля “Временной период между будильниками на встать” и “Временной период между будильниками на выход”. Чтобы время интервалов было больше 0, иначе интервалов не будет, и чтобы они были меньше 20, так как чем меньше интервал, тем меньше вероятность, что пользователь снова уснет. Алгоритм работы подпрограммы Interval показан на рисунке 22.

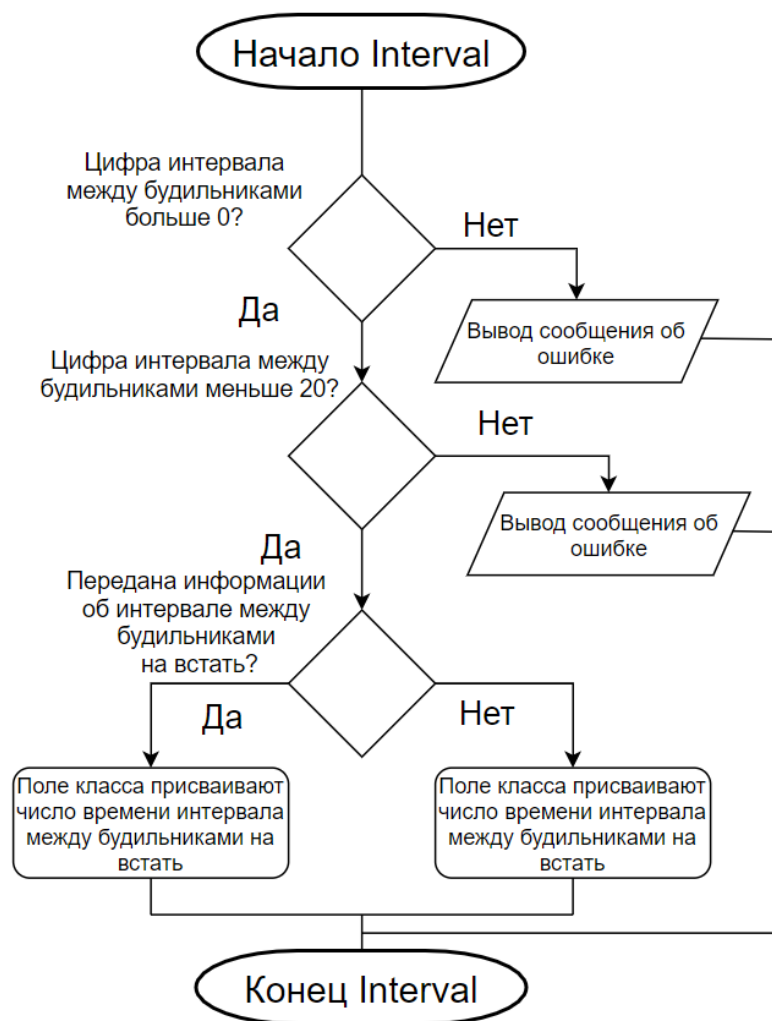


Рисунок 22 – Алгоритм работы подпрограммы HowMany

Схема алгоритма по формированию будильников для обычном расписания показана на рисунке 23.

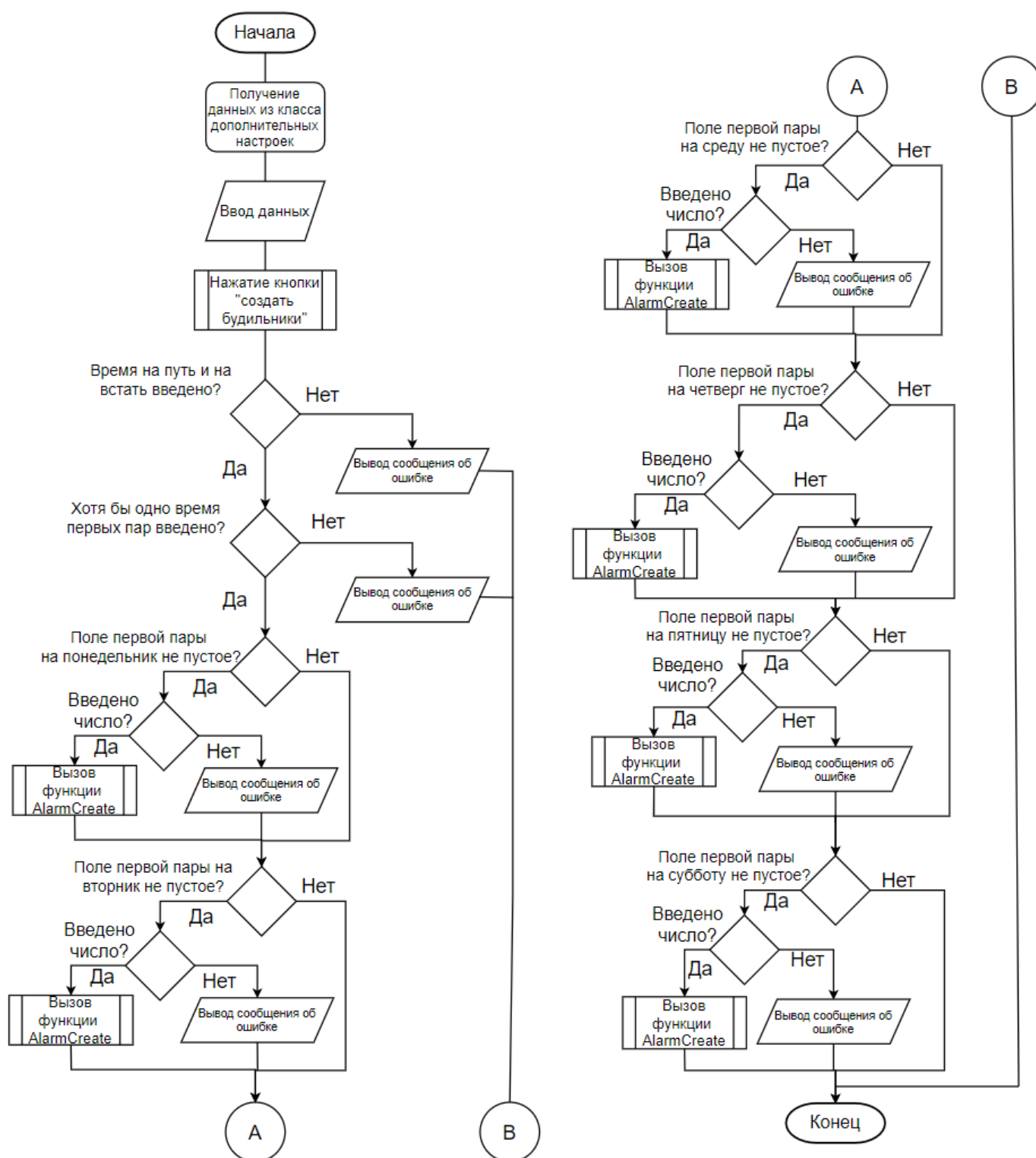


Рисунок 23 – Схема алгоритма по создания будильников для обычном расписания

Подпрограмма AlarmCreate отвечает за финальное создание будильников. Подпрограмма принимает время первой пары, часы и минуты на сборы и на выход, выбранный день, число необходимых будильников и заданные интервалы. Сначала она проверяет, что указанное время введено корректно, иначе высвечивается сообщение об ошибке. Далее, проверка, что введенное число не менее 8 часов. После производится

расчет времени для будильников на выход и далее расчет времени для будильников на встать. И как итог создание заданных будильников. Алгоритм показан на рисунке 24.

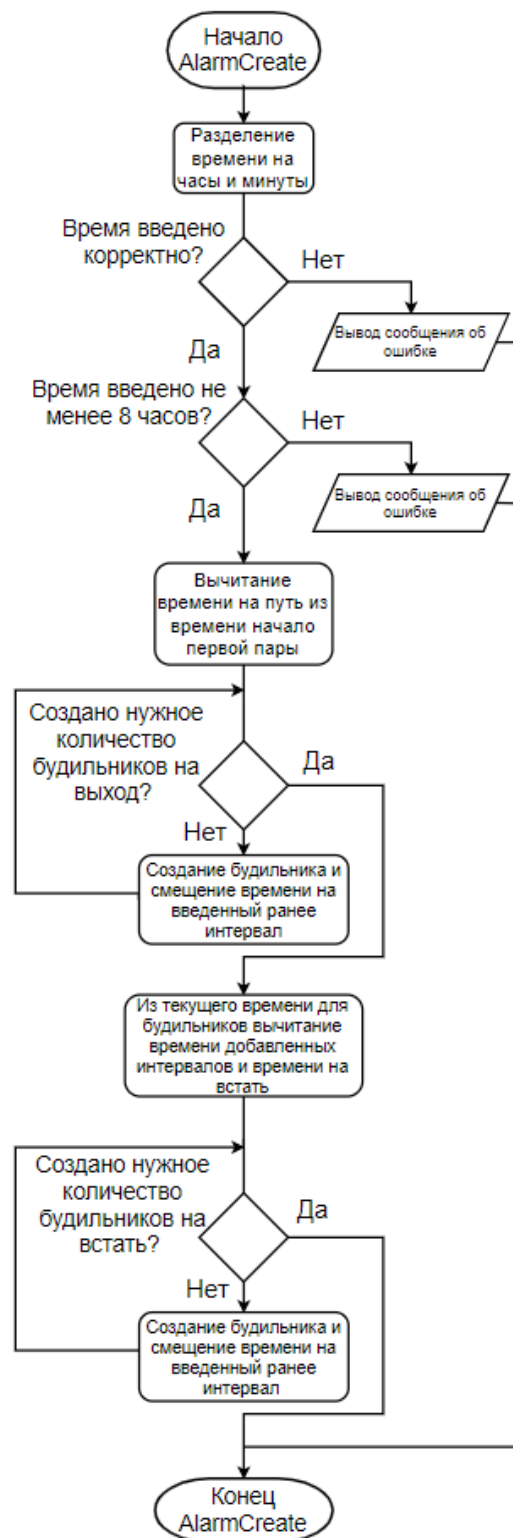


Рисунок 24 – Схема алгоритма работы подпрограммы AlarmCreate

Схема алгоритма по формированию будильников для студентов МГТУ показана на рисунке 25.

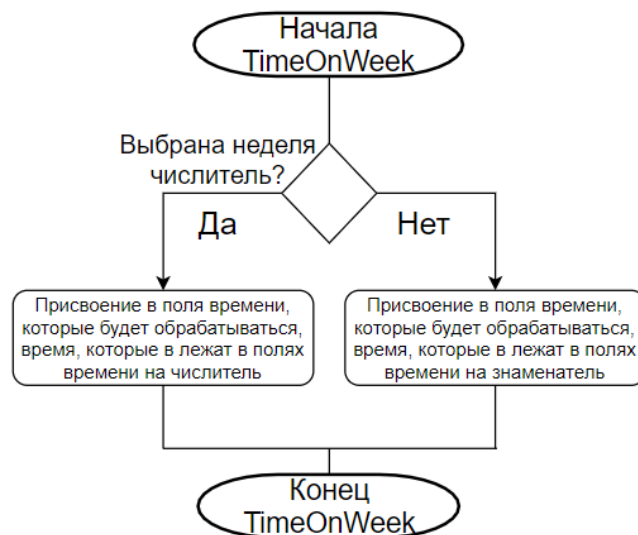


Рисунок 26 – Схема алгоритма подпрограммы TimeOnWeek

Подпрограмма `CreateParametrsAlarn`, сначала проверяет, что время хотя бы на один день введено, потом поочередная проверка ввода времени на каждый день, если все корректно, то подпрограмма передает все ранее введенные параметры подпрограмме `CreateParametrsAlarn`, которая уже и создает будильники. Алгоритм `CreateParametrsAlarn` показан на рисунке 27.

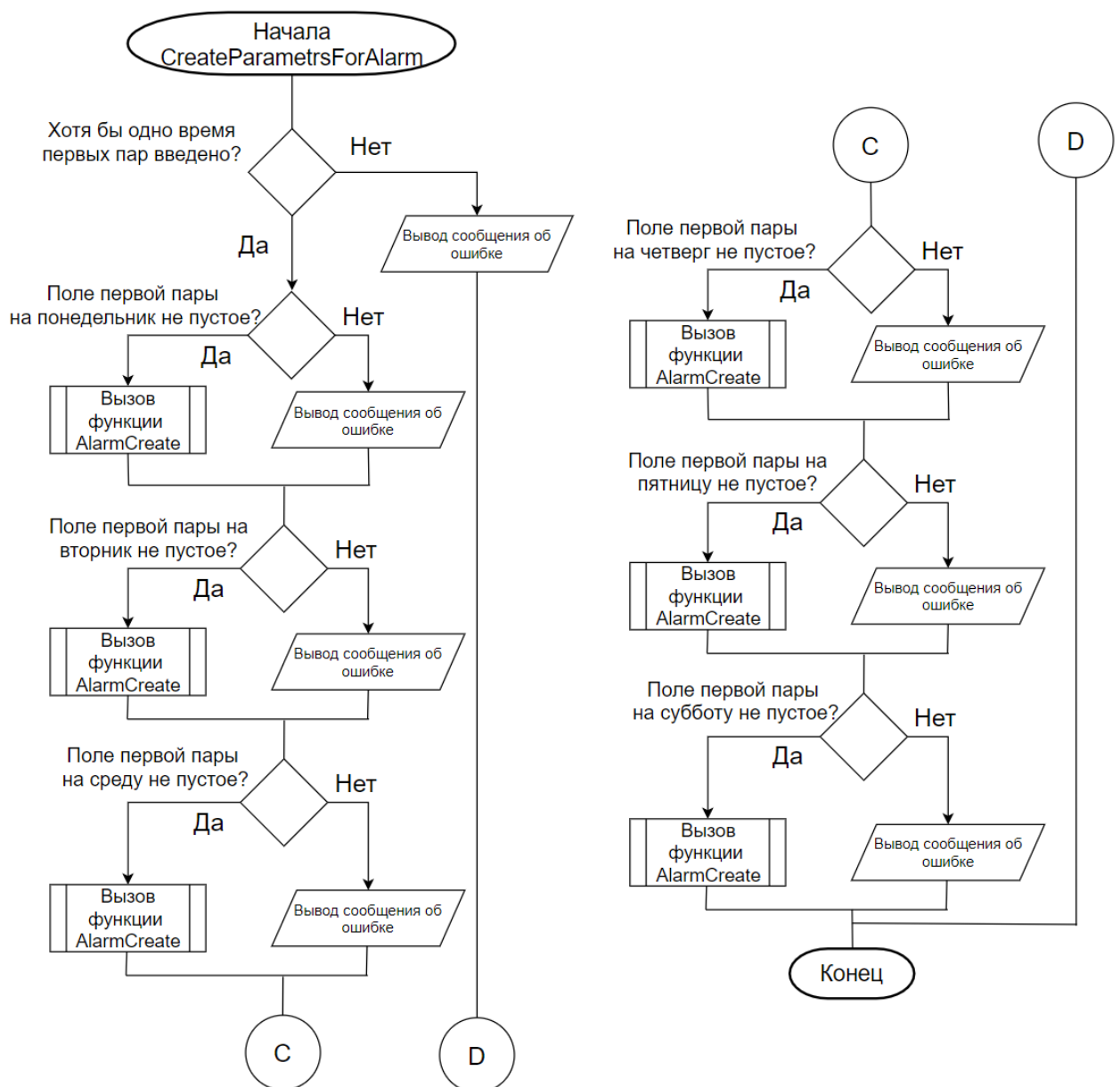


Рисунок 27 – Схема алгоритма подпрограммы CreateParamsAlarm

Подпрограмма AlarmCreate для расписания МГТУ работает почти идентична одноименной подпрограмме для обычного расписания, алгоритм которой показан на рисунке 19, но отличия состоят в том, что в AlarmCreate для расписания МГТУ нет проверки на некорректный ввод времени и что число не менее 8 часов, это делается в другой части программы, алгоритм которой можно увидеть на рисунке 25.

2.5. Разработка диаграммы компоновки

Для более детального понимания связи компонентов и работы приложения в целом, была разработана диаграмма компоновки о связи файлов, показанная на рисунке 28.

Исполняемый apk собирается из исполняющих файлов с расширением .java (SettingOtherTime, Instruction, ChangeWeek, MainActivity, MgtuBaumanka и TimeFirstLesson) и вспоминаящими файлами с расширениями .xml (activity_setting_other_time, activity_instruction, activity_change_week, activity_main, activity_mgtu_baumanka и activity_time_first_lesson), которые содержат визуальную обертку файлов .java [3].

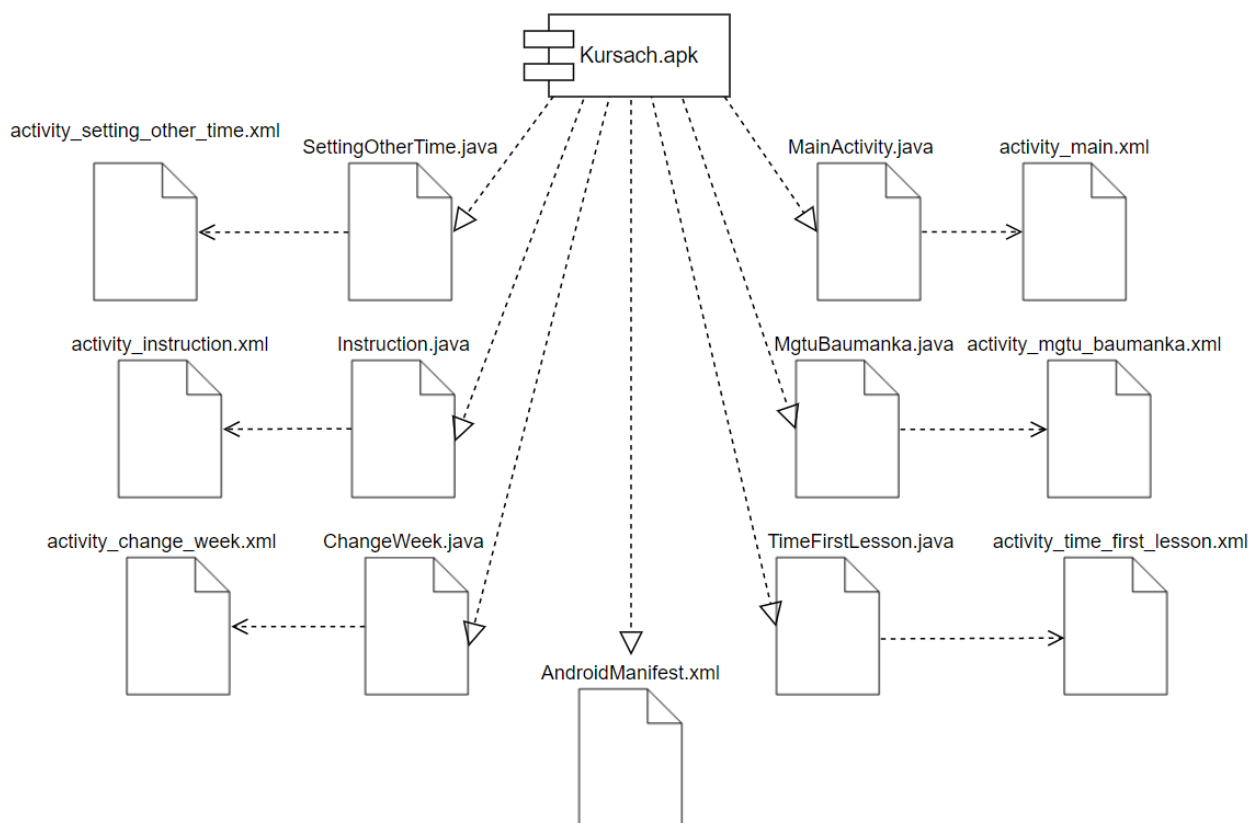


Рисунок 28 – Диаграмма компоновки

3. Выбор стратегии тестирования и разработка тестов

3.1 Тестирование структурным контролем

Одним из видов тестирования был выбран структурный метод, так как он сформирован на продолжительном опыте работы программистов. Разработчики и тестирующие на протяжении долгого времени пополняли перечень вопросов, которые могут выявлять ошибки, что часто встречаются, но в коде все равно могут быть не видны. Результаты тестирования продемонстрированы на таблице 7 [5].

Таблица 7 – тестирование структурным контролем

Вопрос	Результат проверки	Вывод
Обращения к данным		
Все ли переменные инициализированы?	Да, выбранная IDE подсвечивала ошибку, если бы были не инициализированные переменные.	Все переменные инициализированы.
Не превышены ли максимальные (или реальные) размеры массивов и строк?	Используется класс динамического массива для установки дня недели. На данные, которые запоминаются в переменные строкового типа, наложены ограничения на ввод.	Размеры массивов и строк не превышены.
Присутствуют ли переменные со сходными именами?	Нет, выбранная IDE подсвечивала ошибку, если бы были переменные с одинаковыми именами.	Все переменные имеют разные имена.
При вводе из файла проверяется ли завершение файла?	Класс <code>getPreferences</code> автоматически закрывает файл, когда с ним нет работы.	Работа с файлом завершается корректно.
Соответствуют ли типы записываемых и читаемых значений?	Записываются <code>String</code> и считываются <code>String</code> .	Работа с данными в файле в порядке.

Продолжение таблицы 7

Вопрос	Результат проверки	Вывод
Соответствуют ли типы переменных при "наложении" формата в массивах?	Да, массивы Integer-ого типа запоминают данные int-ого типа.	Все типы переменных соответствуют при "наложении".
Не выходят ли индексы за границы массивов?	Класс массива имеет динамический тип и массив создается ровно такой длинны, какой задает программа.	За границу массивов программа не выходит.
Вычисления		
Правильно ли записаны выражения (порядок следования операторов)?	Все процессы проходят последовательно, как ранее было показано на схемах алгоритма.	Сбоев в порядке выполнения операторов нет.
Корректно ли производятся вычисления неарифметических переменных?	Работа с файлом работает корректно	Работа с неарифметическими переменными происходит корректно.
Корректно ли выполнены вычисления с переменными различных типов (в том числе с использованием целочисленной арифметики)?	Если было бы смешение типов, IDE подчеркнуло бы это как ошибку.	Все вычисления проходят корректно.
Возможно ли переполнение разрядной сетки или ситуация машинного нуля?	На все вводимые данные наложены ограничения на ввод.	Переполнение разрядной сетки невозможно.
Присутствуют ли сравнения переменных различных типов?	Переменные integer сравниваются только с переменными integer,	Сравнение различных типов отсутствует

	переменные boolean только с true/false.	
--	---	--

Продолжение таблицы 7

Вопрос	Результат проверки	Вывод
Передача управления		
Будут ли корректно завершены циклы?	Будут ли корректно завершены циклы?	Будут ли корректно завершены циклы?
Будет ли завершена программа?	Будет ли завершена программа?	Будет ли завершена программа?
Существуют ли циклы, которые не будут выполняться из-за нарушения условия входа? Корректно ли продолжатся вычисления?	Существуют ли циклы, которые не будут выполняться из-за нарушения условия входа? Корректно ли продолжатся вычисления?	Существуют ли циклы, которые не будут выполняться из-за нарушения условия входа? Корректно ли продолжатся вычисления?
Интерфейс		
Соответствуют ли списки параметров и аргументов по порядку, типу, единицам измерения?	Соответствуют ли списки параметров и аргументов по порядку, типу, единицам измерения?	Соответствуют ли списки параметров и аргументов по порядку, типу, единицам измерения?
Не изменяет ли подпрограмма аргументов, которые не должны изменяться?	Все изменения с данными происходят в локальных переменных, поля класса сохраняют свои значения.	Программа не изменяет больше параметром, чем нужно для корректной работы.
Не происходит ли нарушения области действия глобальных и локальных переменных с одинаковыми именами?	Все переменные названы по-разному.	Все переменные имеют разные имена, ошибок областей видимости нет.

Тестирование структурным контролем показало, что ошибки, распространённые среди программистов, отсутствуют. То есть в целом программа работает корректно. А более мелкие детали и нюансы будут проверяться уже другими видами тестирований.

3.2 Функциональное тестирование

Функциональное тестирование подразумевает под собой тестирование по принципу “черного ящика”. Вводя данные, которые может ввести пользователь во время использования программы. В тестировании используются четыре вида тестовых наборов:

- эквивалентное разбиение;
- анализ граничных значений;
- анализ причинно-следственных связей;
- предположение об ошибке.

Эквивалентное разбиение работает по принципу деление вводных данных на эквивалентные группы, проверяя тем самым одним тестом множество частных тестов. Сначала создаются классы эквивалентности, и после на их основе проводят тесты. Тестирование эквивалентным разбиением показано на таблице 8 [5].

Классы эквивалентности можно разбить на числа и не числа. Числа можно разбить на рациональные и целые. Целые числа можно разбить на отрицательные и положительные.

Таблица 8 – тестирование эквивалентным разбиением

Номер теста	Назначение теста	Значение исходных данных	Ожидаемый результат	Реакция программы	Вывод
1	Проверка на ввод целых чисел в доп. настройках.	Время на сборы: 120 Время на дорогу: 120 Кол-во будильников на встать: 2 Кол-во будильников на встать: 2 Промежуток между буд-ми на встать: 5	Уведомление “корректные данные сохранены”	Уведомление “корректные данные сохранены”	Программа работает верно.

		Промежуток между буд-ми на выход: 5			
--	--	-------------------------------------	--	--	--

Продолжение таблицы 8

Но-м ер теста	Назначение теста	Значение исходных данных	Ожидаемый результат	Реакция программы	Вывод
2	Проверка на ввод целых чисел в обычном расписании.	Время на понедельник: 830 Время на вторник: 830 Время на среду: 830 Время на четверг: 830 Время на пятницу: 830 Время на субботу: 830	Создаст будильники на каждый день (с понедельника по субботу) в соответствии с введенными дополнительными настройками	Созданы будильники на каждый день (с понедельника по субботу) в соответствии с введенными дополнительными настройками	Про-грамма работает верно.
3	Проверка на ввод целых чисел в расписании МГТУ	Время на понедельник: 830 Время на вторник: 830 Время на среду: 830 Время на четверг: 830 Время на пятницу: 830 Время на субботу: 830	Создаст будильники на каждый день (с понедельника по субботу) в соответствии с введенными дополнительными настройками и выбранной неделей	Созданы будильники на каждый день (с понедельника по субботу) в соответствии с введенными дополнительными настройками и выбранной неделей	Про-грамма работает верно.

4	Проверка на ввод целых чисел во вкладке “смена недели”.	Время напоминания: 12	Будильник на воскресенье на 00:12	Создан будильник на воскресенье на 00:12	Про-грамма работает верно.
---	---	-----------------------	-----------------------------------	--	----------------------------

Продолжение таблицы 8

Номер теста	Назначение теста	Значение исходных данных	Ожидаемый результат	Реакция программы	Вывод
5	Проверка на ввод не чисел в доп. настройках.	Время на сборы: qwe Время на дорогу: qwe Кол-во будильников на встать: qwe Кол-во будильников на встать: qwe Промежуток между буд-ми на встать: qwe Промежуток между буд-ми на выход: qwe	6 уведомлений, в местоположениях полей ввода, “Ведите число”	6 уведомлений, в местоположениях полей ввода “Ведите число”	Про-грамма работает верно.
6	Проверка на ввод не чисел в обычном расписании.	Время на понедельник: qwe Время на вторник: qwe Время на среду: qwe Время на четверг: qwe Время на пятницу: qwe Время на субботу: qwe	6 уведомлений, в местоположениях полей ввода, “Ведите число”	6 уведомлений, в местоположениях полей ввода “Ведите число”	Про-грамма работает верно.
7	Проверка на ввод не чисел в расписании МГТУ	Время на понедельник: qwe Время на вторник: qwe Время на среду: qwe Время на четверг: qwe Время на пятницу: qwe	6 уведомлений, в местоположениях полей ввода,	Сообщение об ошибке работы.	Про-грамма работает не верно, ошибка

		Время на субботу: qwe	“Ведите число”		испра-вл ена.
--	--	-----------------------	----------------	--	------------------

Продолжение таблицы 8

Номер теста	Назначение теста	Значение исходных данных	Ожидаемый результат	Реакция программы	Номер теста
8	Проверка на ввод не чисел во вкладке “смена недели”.	Время напоминания:	Сообщение: “Введите число”	Сообщение: “Введите число”	Про-грамма работает верно.
9	Проверка на ввод рациональных чисел в доп. настрой-ках.	Время на сборы: 12,5 Время на дорогу: 12,5 Кол-во будильников на встать: 6,5 Кол-во будильников на встать: 6,5 Промежуток между буд-ми на встать: 6,5 Промежуток между буд-ми на выход: 6,5	6 уведомлений, в местоположениях полей ввода, “Ведите число”	6 уведомлений, в местоположениях полей ввода “Ведите число”	Программа работает верно.
10	Проверка на ввод рациональных чисел в обычном расписании.	Время на понедельник: 830,5 Время на вторник: 830,5 Время на среду: 830,5 Время на четверг: 830,5	6 уведомлений, в местоположениях полей ввода, “Ведите число”	6 уведомлений, в местоположениях полей ввода “Ведите число”	Программа работает верно.

		Время на пятницу: 830,5 Время на субботу: 830,5			
--	--	--	--	--	--

Продолжение таблицы 8

Номер теста	Назначение теста	Значение исходных данных	Ожидаемый результат	Реакция программы	Номер теста
11	Проверка на ввод рациональных чисел в расписании МГТУ	Время на понедельник: 830,5 Время на вторник: 830,5 Время на среду: 830,5 Время на четверг: 830,5 Время на пятницу: 830,5 Время на субботу: 830,5	6 уведомлений, в местоположениях полей ввода, “Ведите число”	Сообщение об ошибке работы.	Программа работает не верно, ошибка исправлена.
12	Проверка на ввод рациональных чисел во вкладке “смена недели”.	Время напоминания: 12,2	Сообщение: “Введите число”	Сообщение: “Введите число”	Программа работает верно.
13	Проверка на ввод целых отрицательных чисел в доп. настройках.	Время на сборы: -20 Время на дорогу: -20 Кол-во будильников на встать: -20 Кол-во будильников на встать: -20	2 уведомления “Введите число”, 2 “введите число > 0” и	2 уведомления “Введите число”, 2 “введите	Программа работает верно.

		Промежуток между буд-ми на встать: -20 Промежуток между буд-ми на выход: -20	2 “Введите время не менее 1 минуты”	число > 0” и 2 “Введите время не менее 1 минуты”	
--	--	---	-------------------------------------	--	--

Продолжение таблицы 8

Номер теста	Назначение теста	Значение исходных данных	Ожидаемый результат	Реакция программы	Номер теста
14	Проверка на ввод целых отрицательных чисел в обычном расписании.	Время на понедельник: -830 Время на вторник: -830 Время на среду: -830 Время на четверг: -830 Время на пятницу: -830 Время на субботу: -830	6 уведомлений, в местоположениях полей ввода, “Введите время не меньше 8 часов”	Созданы будильники на каждый день (с понедельника по субботу) в соответствии с введенными дополнительными настройками.	Программа работает верно.
15	Проверка на ввод целых отрицательных чисел в расписании МГТУ.	Время на понедельник: -830 Время на вторник: -830 Время на среду: -830 Время на четверг: -830 Время на пятницу: -830	6 уведомлений, в местоположениях полей ввода, “Введите время не меньше 8 часов”	Созданы будильники на каждый день (с понедельника по субботу) в соответствии с введенными дополнительными настройками и	Программа работает не верно, ошибка исправлена.

		Время на субботу: -830		выбранной неделей.	
--	--	---------------------------	--	-----------------------	--

Продолжение таблицы 8

Номер теста	Назначение теста	Значение исходных данных	Ожидаемый результат	Реакция программы	Номер теста
16	Проверка на ввод целых отрицательных чисел во вкладке “смена недели”.	Время напоминания: -12	Сообщение об ошибке	Создает будильник на воскресенье на 23:48	Про-грамма работает верно.

Тестирование эквивалентным разбиением позволило выявить 3 детальных ошибки в программе, которые были не учтены при форматировании логики программы, а также и просто ошибки в коде. Найденные и исправленные ошибки:

- ввод букв в расписание МГТУ;
- ввод рациональных чисел в расписание МГТУ;
- отрицательные числа в будильнике напоминании о смене недели.

Также тестирование черным ящиком включает в себя проверку на граничных условиях. Результаты такого тестирования показаны на таблице 9.

Таблица 9 – тестирование граничных условий

Номер теста	Назначен ие теста	Значение исходных данных	Ожидаемый результат	Реакция программы	Вывод
1	Проверка на ввод 0 во вкладке настройка	Время на сборы: 0 Время на дорогу: 0 Кол-во будильников на встать: 0	2 уведомления “введите положитель-ное число”, 2	2 уведомления “введите положитель-ное число”,	Программа работает верно.

	х доп. времени	Кол-во будильников на встать: 0 Промежуток между буд-ми на встать: 0 Промежуток между буд-ми на выход: 0	“Введите число > 0” и 2 “Введите время не менее 1 минуты”	2 “Введите число > 0” и 2 “Введите время не менее 1 минуты”	
--	-------------------	---	--	--	--

Продолжение таблицы 9

Номер теста	Назначен ие теста	Значение исходных данных	Ожидаемый результат	Реакция программы	Вывод
2	Проверка на ввод 0 во вкладке смена недели	Время напоминания: 0	Уведомление “Введите положитель-но е число”	Уведомление “Введите положитель-но е число”	Програм ма работает верно.
3	Проверка на ввод 0 во вкладке обычного расписани я	Время на понедельник: 0 Время на вторник: 0 Время на среду: 0 Время на четверг: 0 Время на пятницу: 0 Время на субботу: 0	6 уведомлений, в местополо-жен иях полей ввода, “Введите время не меньше 8 часов”	6 уведомле-ний, в местополо-жен иях полей ввода “Введите время не меньше 8 часов”	Програм ма работает верно.
4	Проверка на ввод 0 во вкладке расписа-н ия МГТУ	Время на понедельник: 0 Время на вторник: 0 Время на среду: 0 Время на четверг: 0 Время на пятницу: 0	6 уведомлений, в местополо-жен иях полей ввода, “Введите время не меньше 8 часов”	6 уведомле-ний, в местопол-ожен иях полей ввода “Введите время не	Програм ма работает верно.

		Время на субботу: 0		меньше 8 часов”	
5	Проверка на ввод большого числа во вкладке смена недели	Время напоминания: 999999	Уведомление “Число не соответствует временным размерностям”	Уведомление “Число не соответствует временным размерностям”	Программа работает верно.

Продолжение таблицы 9

Номер теста	Назначение теста	Значение исходных данных	Ожидаемый результат	Реакция программы	Вывод
6	Проверка на ввод большого числа во вкладке настрой-ка х доп. времени	Время на сборы: 999999 Время на дорогу: 999999 Кол-во будильников на встать: 999999 Кол-во будильников на встать: 999999 Промежуток между буд-ми на встать: 999999 Промежуток между буд-ми на выход: 999999	2 уведомления “Число не соответствует временным размерностям”, 2 “Введите число не более чем 10” и 2 “Введите время меньше, 60 минут”	2 уведомления “Число не соответствует временным размерностям”, 2 “Введите число не более чем 10” и 2 “Введите время меньше, 60 минут”	Про-грамма работает верно.
7	Проверка на ввод большого числа во вкладке обычного	По очередной ввод числа “99999” в поля времени на понедельник,	Каждый раз уведомление, в местоположен ии поля ввода, “Число не	Уведомление, в местоположении поля ввода, “Число не соответствует	Про-грамма работает верно.

	расписа-ни я	времени на вторник, времени на среду, времени на четверг, времени на пятницу, времени на субботу	соответствует временным размерностям”	временным размерностям”	
--	-----------------	--	---	----------------------------	--

Продолжение таблицы 9

Номер теста	Назначен ие теста	Значение исходных данных	Ожидаемый результат	Реакция программы	Вывод
8	Проверка на ввод большого числа во вкладке расписани я МГТУ	По очередной ввод числа “99999” в поля времени на понедельник, времени на вторник, времени на среду, времени на четверг, времени на пятницу, времени на субботу	Каждый раз уведомление, в местоположен ии поля ввода, “Число не соответствует временным размерностям ”	Уведомление, в местоположении поля ввода, “Число не соответствует временным размерностям”	Про-гра мма работает верно.

Тестирование на граничных условиях показало, что при вводе значений, не предусмотренных программой, программа выдает уведомления об ошибках и работает корректно.

Еще одна стратегия черного ящика включает в себя анализ причинно-следственных связей, которые тоже могут проявить некоторые ошибки. Тесты показаны на таблице 10.

Таблица 10 – анализ причинно-следственных связей

Номер теста	Назначение теста	Значение исходных	Ожидаемый результат	Реакция программы	Вывод
----------------	------------------	----------------------	------------------------	----------------------	-------

		данных			
1	Проверка на ввод пустых данных во вкладке настройках доп. времени	Все поля пустые	Уведомление “Введите число (время на проснуться)” и “Введите число (время на путь)”	Уведомление “Введите число (время на проснуться)” и “Введите число (время на путь)”	Программа работает верно.

Продолжение таблицы 10

Номер теста	Назначение теста	Значение исходных данных	Ожидаемый результат	Реакция программы	Вывод
2	Проверка на ввод пустых данные во вкладке смена недели	Все поля пустые	Уведомление “Поле для значения пустое”	Уведомление “Поле для значения пустое”	Программа работает верно.
3	Проверка на ввод пустых данных во вкладке обычного расписания	Все поля пустые	Уведомление “Не введено ни одной пары”	Уведомление “Не введено ни одной пары”	Программа работает верно.
4	Проверка на ввод пустых данных во вкладке расписания МГТУ	Все поля пустые	Уведомление “Не введено ни одной пары”	Уведомление “Не введено ни одной пары”	Программа работает верно.

5	Работа кнопки “Закрыть приложение”	—	Программа закрывается	Программа закрывается	Программа работает верно.
6	Работа кнопки “Инструкция”	—	Переход во вкладку “инструкция”	Переход во вкладку “инструкция”	Программа работает верно.
7	Работа кнопки “Иное расписание”	—	Переход во вкладку “смена недели”	Переход во вкладку “смена недели”	Программа работает верно.
8	Работа кнопки “Сменить неделю”	—	Смена будильников ЧЗ на ЗН или наоборот	Смена будильников ЧЗ на ЗН или наоборот	Программа работает верно.

Продолжение таблицы 10

Номер теста	Назначение теста	Значение исходных данных	Ожидаемый результат	Реакция программы	Вывод
9	Работа кнопок “Расписание”	—	Переход во вкладку “Обычного расписания”	Переход во вкладку “Обычного расписания”	Программа работает верно.
10	Работа кнопок “Обычное расписание”	—	Переход во вкладку “Обычного расписания”	Переход во вкладку “Обычного расписания”	Программа работает верно.
11	Работа кнопок “Расписание МГТУ”	—	Переход во вкладку “Расписания МГТУ”	Переход во вкладку “Расписания МГТУ”	Программа работает верно.
12	Работа кнопок “Запомнить”	—	Приложение "запоминает" введенные данные	Приложение "запоминает" введенные данные	Программа работает верно.

13	Работа кнопки Switch “ЗН”	–	Выбор ЗН недели	Происходит выбор ЗН недели	Программа работает верно.
14	Работа кнопки Switch “ЧС”	–	Выбор ЧС недели	Происходит выбор ЧС недели	Программа работает верно.
15	Работа кнопок “Создать будильники”	–	Создаются будильники	Создаются будильники	Программа работает верно.
16	Работа кнопок “удалить будильники”	–	Удаляются будильники	Удаляются будильники	Программа работает верно.

Продолжение таблицы 10

Номер теста	Назначение теста	Значение исходных данных	Ожидаемый результат	Реакция программы	Вывод
17	Работа кнопок “Настройка доп. параметров”	–	Переход во вкладку “Настройка дополнительных параметров”	Переход во вкладку “Настройка дополнительных параметров”	Программа работает верно.
18	Работа кнопок “Вернуться на начальный экран”	–	Переход во вкладку главного экрана	Переход во вкладку главного экрана	Программа работает верно.

Тестирование причинно-следственной связью показало, что программа работает корректно.

В результате тестирования методом черного ящика, то есть функциональным тестированием было выявлено 3 детальные ошибки, 2 из которых были не учтены при разработке и 1 ошибка при написании кода.

3.3 Оценочное тестирование

После проведения тестов на работоспособность, приложение можно протестировать с помощью конечных пользователей, которые дадут свою оценку разработанному приложению. В процессе оценочного тестирования участвовали 7 студентов из разных университетов. Участники тестирования оценивали программу по четырем критериям: удобство интерфейса, работоспособность приложения, правильность работы и универсальность. Результаты опроса показано в таблице 11.

Таблица 11 – Оценочное тестирование

№ пользователя	Удобство интерфейса	Работоспособность приложения	Правильность работы	Универсальность
1	5	10	10	9
2	6	10	10	10
3	4	10	10	8

Продолжение таблицы 11

№ пользователя	Удобство интерфейса	Работоспособность приложения	Правильность работы	Универсальность
4	5	10	10	9
5	7	10	10	10
6	4	10	10	7
7	5	10	10	9
Средняя оценка:	5,1	10	10	8,9

Итоги тестирования показали, что работа приложения происходит корректно, но пользователи не сразу могли разобраться в интерфейсе. В дальнейших версиях приложения надо уделить внимание переработке интерфейса, сделав его более интуитивно понятным.

Заключение

В результате выполнения курсовой работы была изучена среда разработки android studio и язык программирования java, и как итог – разработано приложение организации будильников для учащихся, отличительными чертами которого стали подстраивания будильников под учебное расписание.

Так же в процессе разработки были составлены различные диаграммы и схемы для детального понимания работы программы. И использования их во время разработки для правильного написания кода программы и как итог корректной работы приложения.

После получения готовой рабочей программы было проведено тестирование для проверки правильной и безошибочной работы приложения.

Итоговое приложение соответствует заявленному ТЗ и выполняет все требования. В дальнейшем развитии проекта планируется изменить визуальный интерфейс, сделав более понятным и простым.

Список используемых источников

- 1 Эккель Б. «Философия Java. Библиотека программиста» – М.: Питер, 2009. 640 с.
- 2 Документация по Android Studio [Электронный ресурс]. URL: <https://developer.android.com/> (дата обращения: 26.10.2020)
- 3 Иванова, Г.С. Технология программирования: Учебник для вузов – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. 238 с.
- 4 Стюарт К., Филлипс Билл. Android. Программирование для профессионалов, 3-е издание. – М.: Питер, 2017. 688 с.
- 5 Иванова Г.С., Ничушкина Т.Н. Тестирование программного обеспечения: Методические указания по выполнению лабораторной работы по дисциплине "Технология разработки программных систем" [Текст]. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2019. – 13 с.

Приложение А

Техническое задание