Оглавление

[**Введение** 1](#_Toc73193663)

[**1 Разработка клиентского приложения** 2](#_Toc73193664)

[**1.1 Разработка технического задания** 2](#_Toc73193665)

[*1.1.1 Полное наименование системы и её условное обозначение* 2](#_Toc73193666)

[*1.1.2 Основание для разработки* 2](#_Toc73193667)

[*1.1.3 Назначение и цели создания системы* 3](#_Toc73193668)

[**1.2 Анализ технического задания** 3](#_Toc73193669)

[*1.2.1 Общие положения* 3](#_Toc73193670)

[*1.2.2 Требования к системе* 3](#_Toc73193671)

[**1.3 Стандарты** 4](#_Toc73193672)

[**1.4 Требования к техническим средствам** 5](#_Toc73193673)

[**2 Разработка модели предметной области** 6](#_Toc73193674)

[**2.1 Анализ предметной области** 6](#_Toc73193675)

[**2.2 Разработка структуры классов** 7](#_Toc73193676)

[**3 Разработка структуры приложения** 8](#_Toc73193677)

[**3.1 Разработка архитектуры** 8](#_Toc73193678)

[**3.2 Проектирование пользовательского интерфейса и взаимодействие с ним** 8](#_Toc73193679)

[**Заключение** 11](#_Toc73193680)

[**Список литературы** 12](#_Toc73193681)

[**Приложения** 13](#_Toc73193682)

**Введение**

В данной пояснительной записке рассматривается описание игры «Balls Crew» на основе объектно-ориентированного подхода.

При объектно-ориентированном подходе программа представляет собой описание объектов, их свойств (или атрибутов), совокупностей (или классов), отношений между ними, способов их взаимодействия и операций над объектами (или методов).

Несомненным преимуществом данного подхода является концептуальная близость к предметной области произвольной структуры и назначения. Механизм наследования атрибутов и методов позволяет строить производные понятия на основе базовых и таким образом создавать модель сколь угодно сложной предметной области с заданными свойствами.

Еще одним теоретически интересным и практически важным свойством объектно-ориентированного подхода является поддержка механизма обработки событий, которые изменяют атрибуты объектов и моделируют их взаимодействие в предметной области.

В отличие от других подходов к программированию, объектно-ориентированный подход требует глубокого понимания основных принципов, или концепций, на которых он базируется.

К числу основополагающих понятий ООП обычно относят абстракцию данных, наследование, инкапсуляцию и полиморфизм.

Объектно-ориентированное программирование в настоящее время является абсолютным лидером в области прикладного программирования.

В качестве основного инструмента разработки применяется Intellij IDEA. Язык программирования Java.

**1 Разработка клиентского приложения**

**1.1 Разработка технического задания**

*1.1.1 Полное наименование системы и её условное обозначение*

Полное наименование: Разработка игры «Balls Crew» для проведения увлекательно-развивающего досуга.

Условное обозначение системы: «Balls Crew».

*1.1.2 Основание для разработки*

Основанием для разработки данной информационной системы является приказ по НГТУ на курсовое проектирование по дисциплине «Программирование на языке Java».

*1.1.3 Назначение и цели создания системы*

1.1.3.1 Назначение системы

Программный продукт предназначен для проведения досуга.

1.1.3.2 Цели создания системы

Программный продукт разрабатывается с целью:

1) увлекательно скоротать свободное время;

2) развития реакции. С помощью приложения потребитель сможет весело скоротать время.

1.1.3.3 Основные задачи разработки

1) Обеспечить вход пользователя в игру.

2) Обеспечить проведение игры.

**1.2 Анализ технического задания**

*1.2.1 Общие положения*

Согласно техническому заданию необходимо разработать клиентское приложение, которое предназначено для увлекательного времяпровождения.

*1.2.2 Требования к системе*

1.2.2.1 Требования к структуре и функционированию системы

Программный продукт, разрабатываемый в рамках курсового проекта, должен удовлетворять следующему перечню функциональных требований:

1. Удобный графический интерфейс
2. Рабочая игра
3. Возможность выйти и начать заново
4. Сохранение текущего результата

1.2.2.2 Входные данные

Входными данными при работе с программным продуктом должны быть нажатия пробел.

1.2.2.3 Выходные данные

Выходными данными при работе программы являются данные, выводимые на экран пользователя.

1.2.2.4 Требования к надёжности

Система должна сохранять работоспособность и обеспечивать восстановление своих функций при возникновении внештатных ситуаций.

1.2.2.5 Требования к эргономике и технической эстетике

Подсистемы ввода данных, а также формирования и визуализации отчетности должны обеспечивать удобный для конечного пользователя интерфейс.

Главное окно программного продукта должно позволять пользователю начать игру.

Остальные окна программного продукта содержат игровое поле, окно проигрыша и информацию об авторе.

**1.3 Стандарты**

Программный продукт разрабатывается на основании следующих государственных стандартов:

1. 2.103-68 ЕСКД. Стадии разработок
2. 2.104-68 ЕСКД. Основные надписи
3. 2.105-95 ЕСКД. Общие требования к текстовым документам
   1. -96 ЕСКД. Текстовые документы
4. 2.111- 68 ЕСКД. Общие требования к текстовым документам.
5. 2.118-73 ЕСКД. Техническое предложение
6. 2.120-73 ЕСКД. Технический проект
7. 2.316-68 ЕСКД. Правила нанесения на чертежах надписей, технических требований и таблиц
8. 7.1-2003 Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления.

**1.4 Требования к техническим средствам**

Для удобства работы система должна обеспечивать отображение GUI с расширенным набором пользовательских элементов, что соответствует платформе Java, IntelliJ IDEA не ниже версии 8.0.2. ОС не важна, главное-виртуальная машина Java выше представленной версии JRE

Минимальные технические характеристики компьютера, на котором гарантируется стабильная работа программы:

1. компьютер/процессор: компьютер-терминал с процессором класса Intel Celeron 420 с тактовой частотой 1,6 ГГц и старше;
2. память: 512 МБ ОЗУ;
3. монитор: монитор с разрешением 800x600 точек или более высоким, поддерживающий 32000 цветов;
4. наличие свободного дискового пространства на жёстком диске-не менее 512 Мб.

**2 Разработка модели предметной области**

**2.1 Анализ предметной области**

Программируемое приложение должно позволять пользователю свободно управлять персонажем и его действиями.

Нефункциональные требования к программному средству:

1. надежность: программа должна быть автономной;
2. эффективность: программа должна иметь минимальные требования к аппаратному обеспечению. Не должно требоваться дополнительных периферийных средств (сканер, дополнительные дисководы и т.д.)
3. изучаемость: программа должна быть интуитивно понятна, иметь удобный пользовательский интерфейс;

**2.2 Разработка структуры классов**

В соответствие с анализом предметной области можно выделить основные классы информационной системы и их атрибуты:

1. Класс BasicController описывает общие для контроллеров поля

* Список окон приложения
* Экземпляр класса, управляющий отображением

1. Класс GameFormController описывает основной функционал игрового окна

* Кнопка перехода в меню
* Кнопка старта/рестарта игры
* Передача действий пользователя в игровой движок
* Инициализация игрового движка

1. Класс GameFormHardController описывает функционал, аналогичный GameFormController, но для игры повышенной сложности
2. Класс MainFormController описывает функционал главного меню приложения
   * Начало обычной игры
   * Начало усложненной игры
3. Класс GameConfig описывает интерфейс для конфигурационных классов игры
   * Доступные препятствия
   * Доступные элементы окружения
   * Экземпляр класса управляемого игроком объекта
   * Радиус, в котором будут появляться препятствия
   * Объект, на котором будет происходить отрисовка
   * Количество одновременно существующих препятствий
4. Классы EasyGameConfig и HardGameConfig содержат конфигурацию для простого и усложненного режимов игры соответственно
5. Класс GameEngine описывает процесс игры. Его функциональность:
   * Запуск, остановка и перезапуск игры
   * Передача пользовательского ввода сервисам
   * Инициализация и контроль сервисов
   * Инициализация объектов
6. Класс ObjectHub предоставляет контроль над игровыми объектами
   * Хранение созданных объектов
   * Хранения доступных для создания объектов
7. Класс BasicCyclicService описывает класс потока сервиса. Он позволяет
   * Остановить/запустить/перезапустить сервис
8. Класс CollisionCyclicService описывает поведение игры при столкновении объектов
9. Класс DrawCyclicService выполняет отрисовку объектов
10. Класс MoveCyclicService выполняет движение объектов
11. Класс RespawnCyclicService выполняет респавн объектов
12. Класс ScopeCyclicService выполняет обновление счетчика счета игрока
13. Абстрактный класс BasicObject описывает базовые свойства объекта
14. Интерфейс Collided описывает функциональность для сталкиваемых объектов
15. Интерфейс Drawable описывает функциональность для отображаемых объектов
16. Интерфейс Movable описывает функциональность для передвигаемых объектов
17. Интерфейс PlayerControlled описывает функциональность для объектов, реагирующих на ввод пользователя
18. Интерфейс Respawned описывает функциональность для пересоздаваемых объектов
19. Абстрактный класс SimpleFigureObject описывает основные свойства для объектов – фигур
20. Класс Actor описывает объект, управляемый игроком
21. Классы Circle, Cube, EnemyCircle, Rectangle описывают типовые препятствия
22. Класс Floor описывает объект окружения – пол
23. Интерфейс Move описывает функциональность для поведения объекта – движения
    * Передвижение объекта
    * Активация/деактивация/обновление поведения
24. Классы JumpMove, LeftMove, RightMove описывают поведение, активируемое действиями игрока
25. Классы RandomJumpMove и RunImitationMove описывают автономное поведение
26. Класс Main инициализирует главное окно приложения и загружает шаблоны окон приложения
27. Ресурсы GameForm, GameFormHard и MainForm шаблоны окон приложения, обрабатываемые фремворком JavaFX

**3 Разработка структуры приложения**

**3.1 Разработка архитектуры**

Так как «Balls Crew», программа относительно не сложная, нам понадобятся несколько классов с описаниями различных форм и различных объектов на них.

**3.2 Проектирование пользовательского интерфейса и взаимодействие с ним**

В данном проекте для продуктивной работы реализован удобный пользовательский интерфейс, похожий на большинство Windows – приложений (рис. 1).

Рисунок 1 – Интерфейс меню приложения.

Рисунок 2 – Основной интерфейс приложения.

Рисунок 3 – Интерфейс окна проигрыша.

Рисунок 4 – Интерфейс окна «Об авторе».

Основной сценарий работы пользователя с приложением:

1. Пользователь нажимает на кнопку «Играть».
2. Пользователь видит препятствия, через которые он должен перепрыгивать.
3. Пользователь рано или поздно врежется в препятствие и увидит меню проигрыша.
4. Пользователь закрывает окно проигрыша или оставляет его и продолжает играть, набирая очки сначала, или закрывает игровую форму.

Рисунок 12 – Диаграмма деятельности.

**Заключение**

В данной курсовой работе была разработана игра «Balls Crew» на основе объектно-ориентированного подхода. Этот метод позволяет обеспечить разделение данных, управление доступом к ним, а также расширяемость и модифицируемость.

В проекте удалось реализовать все поставленные задачи:

1. Создать графический интерфейс
2. Обеспечить корректную реакцию приложения на действия игрока
3. Обеспечить выход из приложения и начала новой игры

Однако, остается широкое поле для дальнейшей деятельности по развитию и улучшению проекта.

Таким образом, следует продолжить работу над проектом в будущем и выполнить определенные в ходе анализа проведенной деятельности задачи по доработке программы.

**Список литературы**

1. <https://stackoverflow.com>
2. <https://www.youtube.com>

**Приложения**