

ВВЕДЕНИЕ

Цель –разработка программы, которая создает структуру игрового холдинга, аналогичную платформе Steam. Программа должна предусматривать создание записей для каждой игры отдельно и для каждого пользователя, включая коллекцию их игр.

Задачи:

- Составить IDEF0-диаграмму проекта;
- Составить блок-схемы алгоритмов;
- Реализовать алгоритмы на языке Object Pascal;
- Протестировать реализации алгоритмов.

Аналитическая часть

Структура представлена на рисунке 1 в виде IDEF0-нотации. Задача разделяется на несколько блоков, представленных на рисунке 2:

- Ввод данных с клавиатуры;
- Выполнение произведения;
- Формирование коллекций игр;
- Вывод данных на экран.

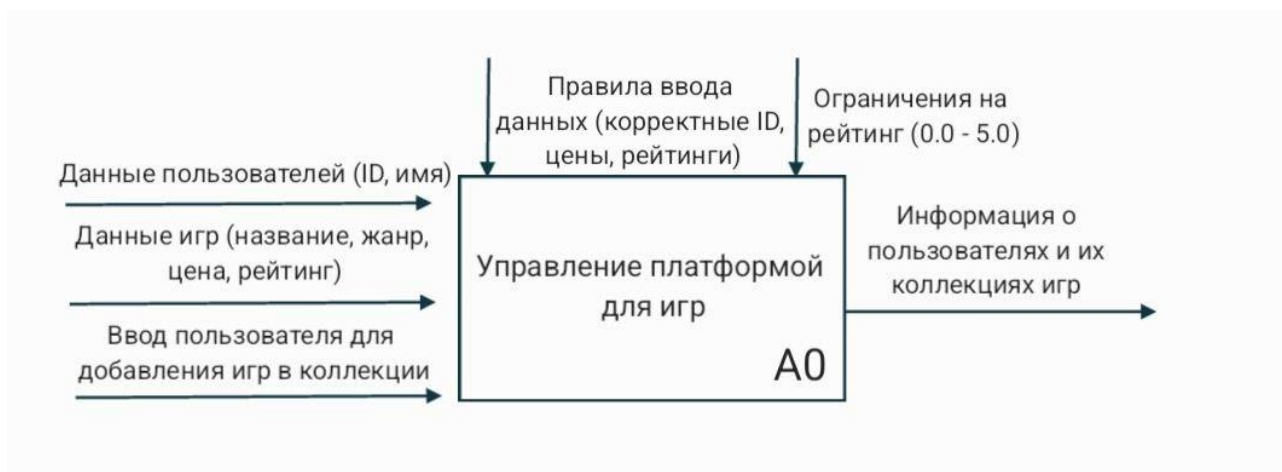


Рисунок 1 – Общая IDEF0-нотация

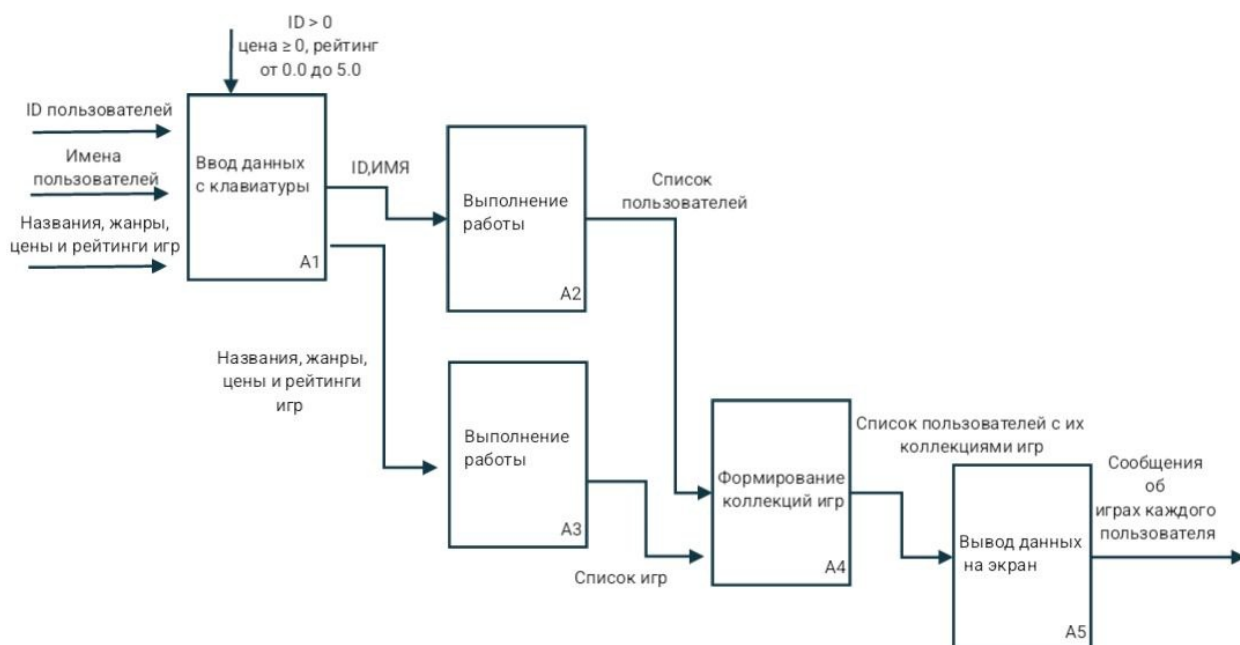


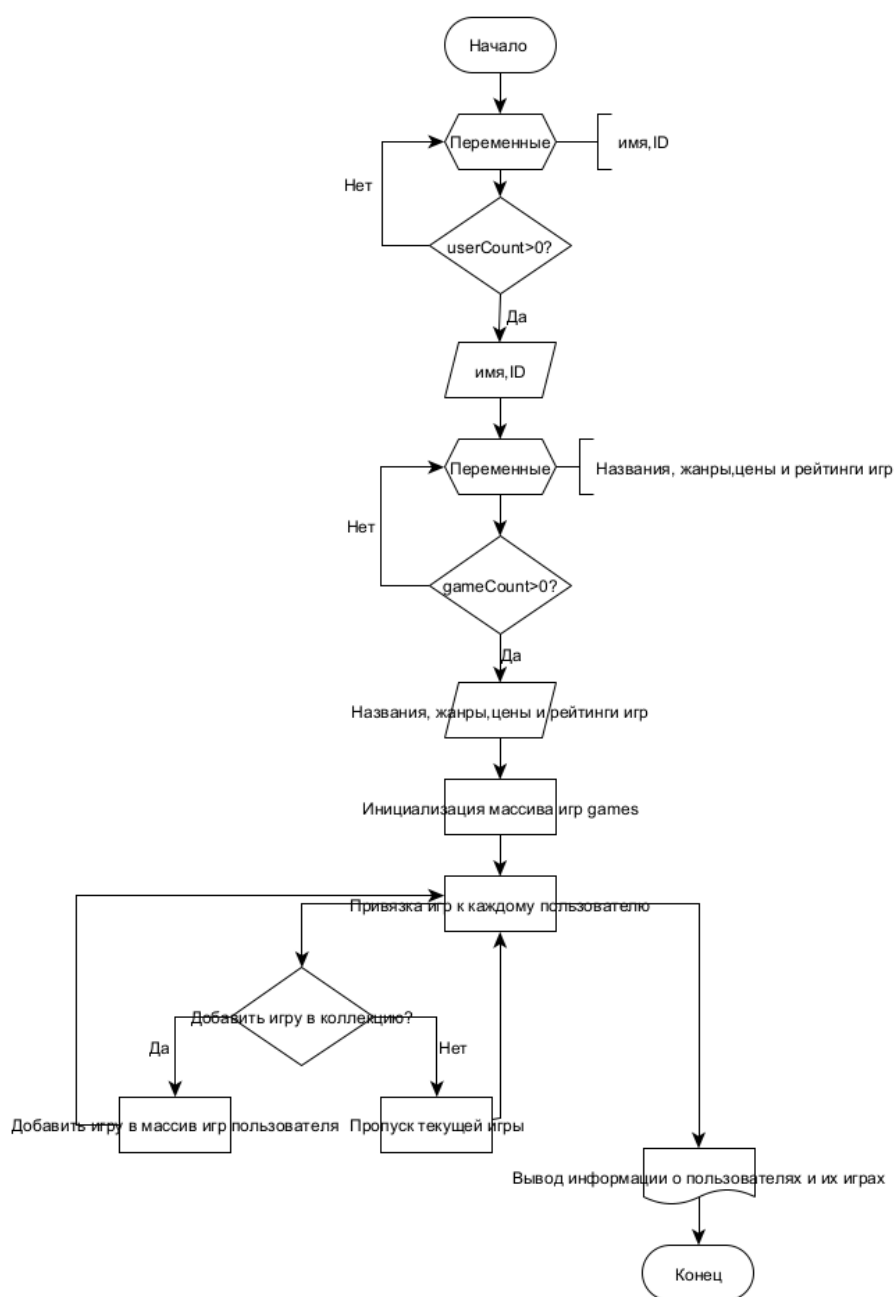
Рисунок 2 – Подробная IDEF0-нотация

Конструкторская часть

Блоки IDEF0-диаграммы представляют собой 1-2 действия, ввиду чего рациональнее отобразить алгоритм всего проекта целиком без разбиения каждого блока на отдельные процедуры.

Блок-схема алгоритма представлена на рисунке 1. Элементы, отвечающие за интерфейс пользователя, на блок-схеме не отображены; текстовые сообщения, ввиду малозначимости их дословного приведения, представлены сокращенно.

Рисунок 1 – Блок-схема алгоритма программы



Технологическая часть

Реализация алгоритма

В настоящем разделе представлена реализация алгоритма, чья блок-схема представлена на рисунке 1. Реализация была произведена с помощью языка программирования Pascal и представлена в листинге 1.

Листинг 1 – Программа алгоритма

```

program GamingPlatform;

type
  Game = record
    name: string;
    genre: string;
    price: real;
    rating: real;
  end;

  User = record
    id: integer;
    name: string;
    games: array of Game;
  end;

var
  users: array of User;
  games: array of Game;
  userCount, gameCount, i, j, answer: integer;
  tempInput: string;

procedure AddGameToUser(var user: User; game: Game);
begin
  SetLength(user.games, Length(user.games) + 1);
  user.games[High(user.games)] := game;
end;

procedure PrintUserInfo(user: User);
var
  i: integer;
begin
  writeln('Пользователь: ', user.name, ' (ID: ', user.id, ')');
  writeln('Коллекция игр:');
  if Length(user.games) = 0 then
    writeln('Нет игр в коллекции.')
  else
    for i := 0 to High(user.games) do
      writeln(' ', user.games[i].name, ' - ', user.games[i].genre, ', ', user.games[i].price:0:2,
        '$, рейтинг: ', user.games[i].rating:0:1);
    end;
end;

begin
  repeat
    write('Введите количество пользователей: ');
    readln(tempInput);
    Val(tempInput, userCount, answer);
    if (answer <= 0) or (userCount <= 0) then
      writeln('Ошибка: введите положительное целое число.');
    until (answer = 0) and (userCount > 0);

    SetLength(users, userCount);

    for i := 0 to userCount - 1 do
      begin
        repeat
          write('Введите ID пользователя (целое число): ');
          readln(tempInput);
          Val(tempInput, users[i].id, answer);
          if (answer <= 0) or (users[i].id <= 0) then
            writeln('Ошибка: ID должен быть положительным числом.');
          until (answer = 0) and (users[i].id > 0);

          write('Введите имя пользователя: ');
          readln(users[i].name);
          SetLength(users[i].games, 0);
        end;

        repeat
          write('Введите количество игр: ');
          readln(tempInput);
          Val(tempInput, gameCount, answer);
          if (answer <= 0) or (gameCount <= 0) then
            writeln('Ошибка: введите положительное целое число.');
          until (answer = 0) and (gameCount > 0);

          SetLength(games, gameCount);

          for i := 0 to gameCount - 1 do
            begin
              writeln('Введите название, жанр, цену и рейтинг игры:');
              write('Название: '); readln(games[i].name);
              write('Жанр: '); readln(games[i].genre);
              repeat
                write('Цена (положительное число): ');
                readln(tempInput);
                Val(tempInput, games[i].price, answer);
                if (answer <= 0) or (games[i].price <= 0) then
                  writeln('Ошибка: цена должна быть положительным числом.');
                until (answer = 0) and (games[i].price >= 0);
              repeat
                write('Рейтинг (0.0 - 5.0): ');
                readln(tempInput);
                Val(tempInput, games[i].rating, answer);
                if (answer <= 0) or (games[i].rating < 0.0) or (games[i].rating > 5.0) then
                  writeln('Ошибка: рейтинг должен быть в диапазоне от 0.0 до 5.0.');
                until (answer = 0) and (games[i].rating >= 0.0) and (games[i].rating <= 5.0);
            end;

            for i := 0 to High(users) do
              begin
                writeln('Игры для пользователя: ', users[i].name);
                for j := 0 to High(games) do
                  begin
                    repeat
                      write('Добавить игру "', games[j].name, '" в коллекцию? (1 - да, 0 - нет): ');
                      readln(tempInput);
                      Val(tempInput, answer, answer);
                      if (answer <= 0) or ((answer <= 1) and (answer <= 0)) then
                        writeln('Ошибка: введите 1 или 0.');
                      until (answer = 1) or (answer = 0);
                      if answer = 1 then
                        AddGameToUser(users[i], games[j]);
                      end;
                    end;

                    writeln('Информация о пользователях:');
                    for i := 0 to High(users) do
                      PrintUserInfo(users[i]);
                    end;
                  end;

                  // Разделение на четные и нечетные элементы
                  evenArr := '';
                  oddArr := '';
                  for i := 0 to High(arr) do
                    if arr[i] mod 2 = 0 then
                      evenArr := evenArr + IntToStr(arr[i]) + ' ';
                    else
                      oddArr := oddArr + IntToStr(arr[i]) + ' ';
                    end;
                  end;

                  // Вывод четных и нечетных элементов
                  writeln('Четные элементы: ', evenArr);
                  writeln('Нечетные элементы: ', oddArr);
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end.

```

Тестирование реализации

Для данной реализации были использованы методы тестирования черным ящиком – метод эквивалентного разбиения – и белым ящиком – метод комбинаторного покрытия условий и решений.

Были выделены следующие эквивалентные классы возможных решений:

1. Корректные данные для пользователей и игр, сообщение о добавлении игры выведено корректно.
2. Нарушение ввода – ввод не целых значений для ID, цены или рейтинга.
3. Нарушение диапазона значений (например, отрицательные цены, рейтинг за пределами диапазона).
4. Отсутствие игр у пользователя.

Результаты тестов представлены в таблице 1. Сообщения о системной ошибке предусмотрены для некорректного ввода данных.

Таблица 1 – Тесты реализации программы

№	Входные данные	Выходные данные
1	Пользователь: ID=1, Имя=Иван; Игра: Doom, FPS, Цена=29.99, Рейтинг=4.8	Пользователь Иван; Коллекция: Doom, FPS, Цена: 29.99, Рейтинг: 4.8
2	Пользователь: ID=2, Имя=Анна; Игра: Stardew Valley, Симулятор, Цена=14.99, Рейтинг=4.9	Пользователь Анна; Коллекция: Stardew Valley, Симулятор, Цена: 14.99, Рейтинг: 4.9
3	ID пользователя="abc", Имя=Олег	Системная ошибка: "Ошибка: ID должен быть положительным числом."
4	Игра: Cyberpunk, RPG, Цена=-10.0, Рейтинг=3.5	Системная ошибка: "Ошибка: цена должна быть положительным числом."
5	Игра: The Sims, Симулятор, Цена=40.0, Рейтинг=5.5	Системная ошибка: "Ошибка: рейтинг должен быть в диапазоне от 0.0 до 5.0."
6	Пользователь: ID=3, Имя=Максим, Игры: отсутствуют	Пользователь Максим; Коллекция игр пуста.

ЗАКЛЮЧЕНИЕ

В настоящей работе была составлена программа для управления игровой платформой. Реализована возможность создания записей для игр и пользователей, добавления игр в коллекции пользователей, а также вывода информации о них.

Задачи:

- Составлена IDEF0-диаграмма проекта;
- Составлены блок-схемы алгоритмов;
- Алгоритмы реализованы на языке Object Pascal;
- Реализации алгоритмов протестированы, были подобраны классы ошибок.