Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**Иркутский национальный исследовательский технический университет**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |
| Кафедра вычислительной техники |
| наименование кафедры |

|  |
| --- |
| **Отчет** |
| по лабораторной работе №4 по дисциплине «Разработка классов с использованием механизмов наследования, полиморфизма и инкапсуляции.» |
| наименование темы  Вариант №4 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент |  | АСУб-19-1 |  |  |  | Аласаев П.М |
|  |  | шифр |  | подпись |  | И.О. Фамилия |
| Проверил |  |  |  |  |  | Маланова Т.В. |
|  |  |  |  | подпись |  | И.О. Фамилия |
| Работа защищена с оценкой | | | |  | | |

Иркутск 2020 г.

**Оглавление**

[1 Постановка задачи 3](#_Toc56894035)

[2 Описание структуры пользовательского меню 4](#_Toc56894036)

[3 Проектирование классов 5](#_Toc56894037)

[4 Таблица тестов 6](#_Toc56894038)

[5 Результаты тестирования 7](#_Toc56894039)

[6 Листинг исходного кода 14](#_Toc56894040)

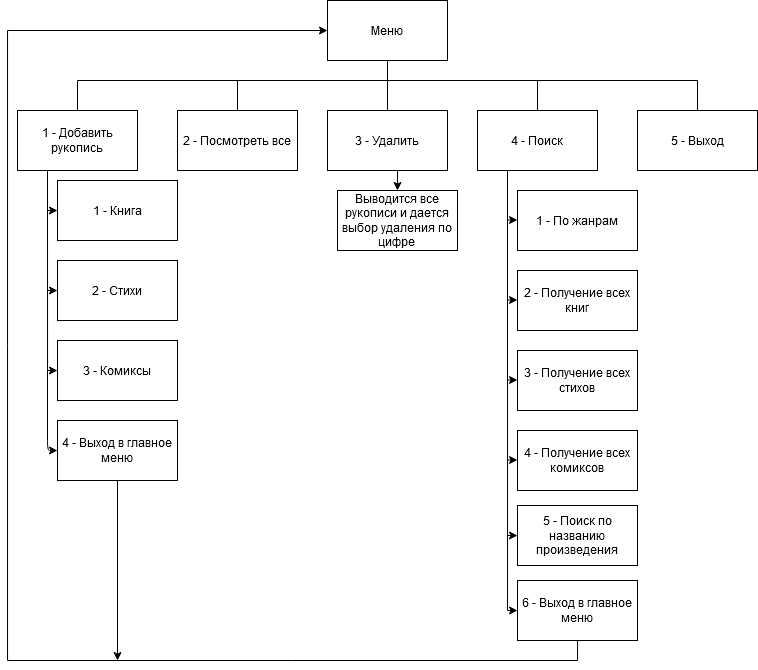
[Список использованных источников 30](#_Toc56894041)

1 Постановка задачи

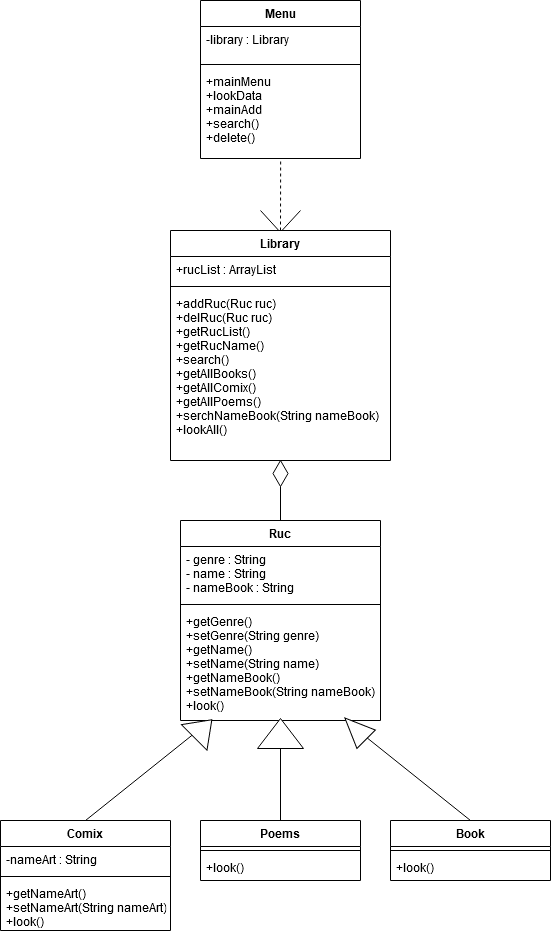
В соответствии с индивидуальным заданием описать иерархию классов, для каждого класса описать поля и соответствующие методы доступа к ним. В зависимости от задания некоторые из этих классов(как минимум один) являются абстрактными и служат для выделения общих данных и поведения для других классов. Абстрактный класс должен содержать как минимум один абстрактный метод, реализация которого у его наследников должна различаться. Помимо этого в общую часть задания входит разработка класса группирующего объекты описанных в соответствии с заданием классов. Для первого варианта индивидуального задания это может быть класс «кафедра». Для выполнения задания необходимо создать некоторое количество объектов, добавить их в группу используя предусмотренные методы класса-«группы» и для каждого из них вызвать унаследованный метод.

Вариант: Рукопись, книга, стихи, комикс.

2 Описание структуры пользовательского меню

  
Рис 1 – Структура пользовательского меню

3 Проектирование классов

  
Рис 2 – Проектирование классов

4 Таблица тестов

Таблица 1 – Таблица тестов

|  |  |  |
| --- | --- | --- |
| № | Описание теста | Результат работы программы |
| 1 | Ввод в меню числа, не существующего пунктам меню | Сообщение об ошибке, повторение данного пункта меню |
| 2 | Ввод в главном меню «1» | Переход в меню «Добавить рукопись» |
| 3 | Ввод в меню Добавить данные «1» | Предлагает ввести данные |
| 4 | Вводим данные | Возвращение в меню добавления данных. |
| 5 | Ввод в меню Добавить данные «2» | Предлагает ввести данные |
| 6 | Вводим данные | Возвращение в меню добавления данных. |
| 7 | Ввод в меню Добавить данные «3» | Предлагает ввести данные |
| 8 | Вводим данные | Возвращение в меню добавления данных. |
| 9 | Ввод в главном меню «2» | Вывод всех рукописей, Выход в главное меню |
| 10 | Ввод в главном меню «3» | Переход в меню «Удалить», выводятся все пронумерованные рукописи и при выборе цифры рукопись удаляется |
| 11 | Ввод в главном меню «4» | Переход в меню «Поиск» |
| 12 | Ввод в меню Поиск «1» | Предлагается ввести жанр |
| 13 | Ввод жанра | Вывод рукописей с таким же жанром |
| 14 | Ввод в меню Поиск «2» | Выводятся все книги, которые записаны |
| 15 | Ввод в меню Поиск «5» | Предлагается ввести название книги |
| 16 | Ввод в главном меню «5» | Выход из программы |

5 Результаты тестирования

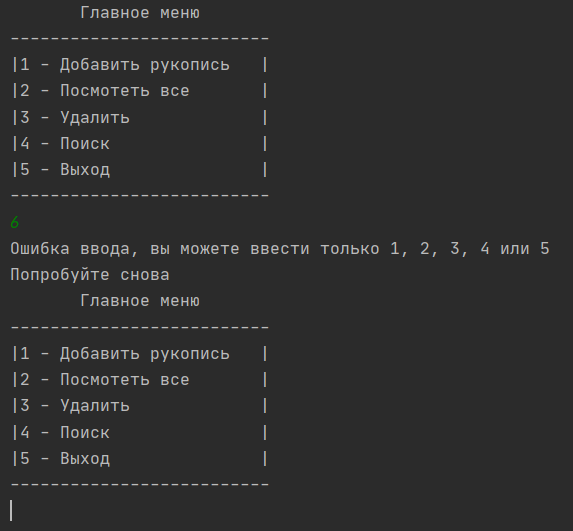


Рис 3 – Тест 1

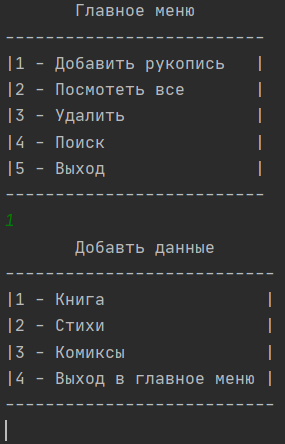


Рис 4 – Тест 2

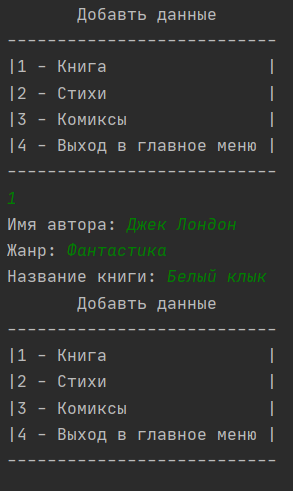


Рис 5 – Тест 3 и 4

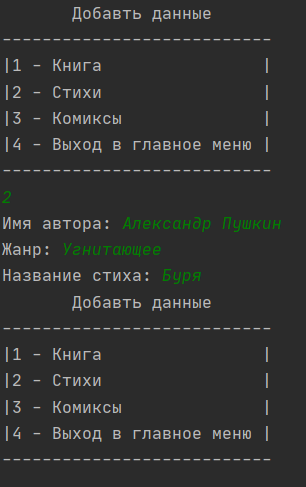


Рис 6 – Тест 5 и 6

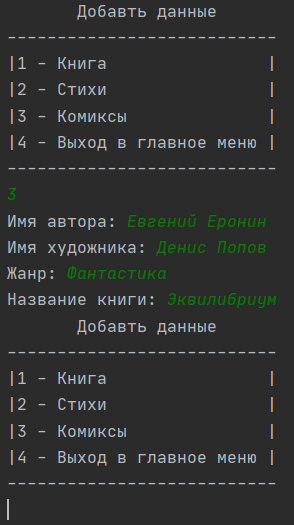


Рис 7 – Тест 7 и 8

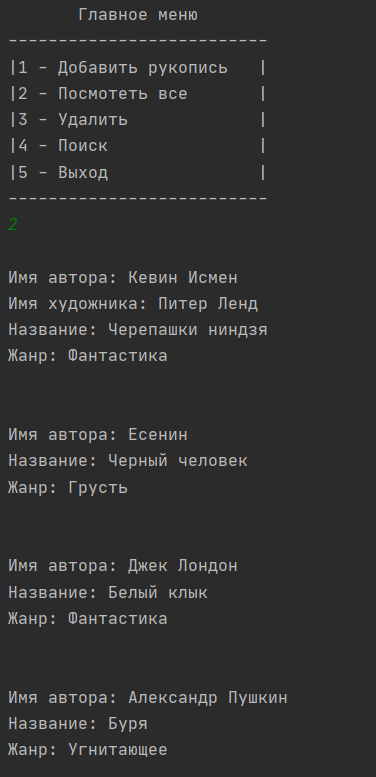


Рис 8 – Тест 9

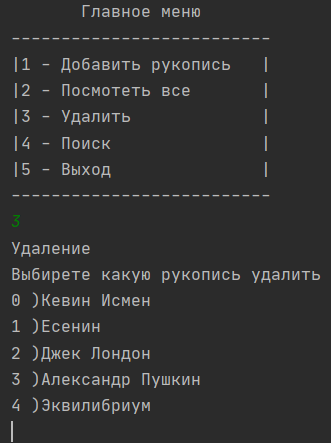


Рис 8 – Тест 10

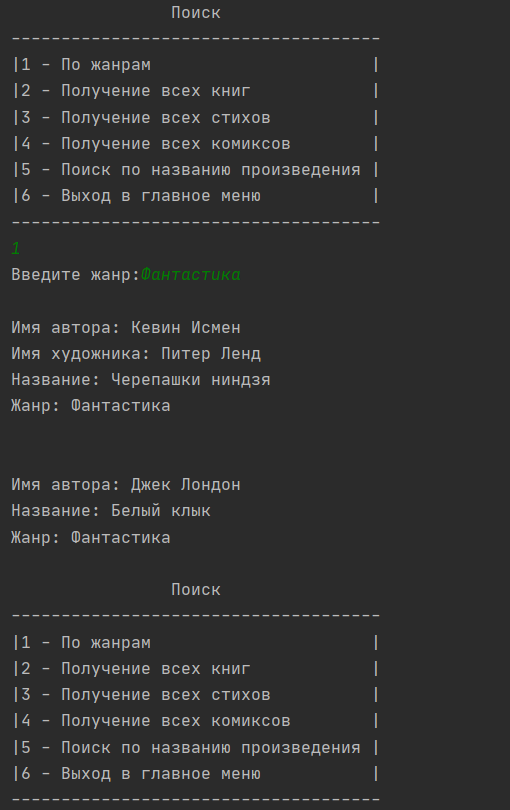


Рис 9 – Тест 11, 12 и 13

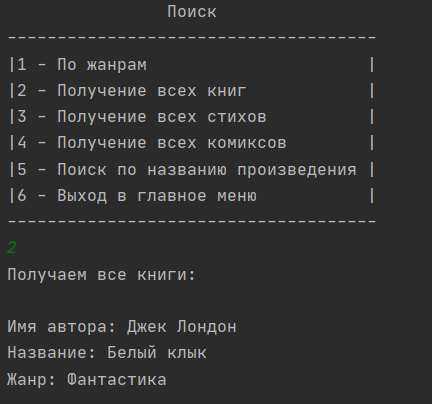


Рис 10 – Тест 14

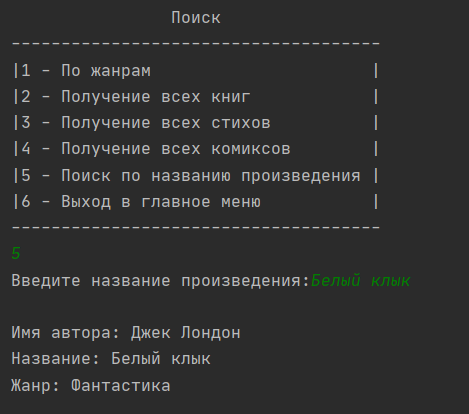


Рис 11 – Тест 15

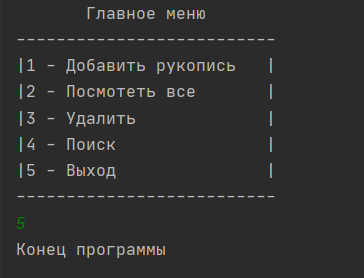


Рис 12 – Тест 16

5 Листинг исходного кода

1) Manu

package lab4.manu;

import lab4.Library;

import lab4.data.Book;

import lab4.data.Comix;

import lab4.data.Poems;

import lab4.data.Ruc;

import java.util.ArrayList;

import java.util.Scanner;

public class Menu {

private static Library library = new Library();

private static Scanner in = new Scanner(System.in);

public static void main (String[] args){

if (library.getRucList().isEmpty()) {

Book republicDragon = new Book("Фантастика", "Ребекка Хуанг", "Республика Дракон");

Comix turtles = new Comix("Питер Ленд","Фантастика", "Кевин Исмен", "Черепашки ниндзя");

Poems blackMan = new Poems("Грусть", "Есенин", "Черный человек");

library.addRuc(republicDragon);

library.addRuc(turtles);

library.addRuc(blackMan);

}

mainMenu();

}

public static void mainMenu(){

int a = 0;

do{

System.out.println(" Главное меню ");

System.out.println("--------------------------");

System.out.println("|1 - Добавить рукопись |");

System.out.println("|2 - Посмотеть все |");

System.out.println("|3 - Удалить |");

System.out.println("|4 - Поиск |");

System.out.println("|5 - Выход |");

System.out.println("--------------------------");

try {

a = Integer.parseInt(in.nextLine());

if (a < 1 || a > 5) throw new IllegalArgumentException();

} catch (IllegalArgumentException e){

System.out.println("Ошибка ввода, вы можете ввести только 1, 2, 3, 4 или 5");

System.out.println("Попробуйте снова");

}

switch (a){

case 1:

mainAdd();

break;

case 2:

lookData();

break;

case 3:

delete();

break;

case 4:

search();

break;

}

}while(a != 5);

System.out.println("Конец программы");

}

public static void lookData(){

library.lookAll();

}

public static void mainAdd(){

int s = 0;

do {

System.out.println(" Добавть данные ");

System.out.println("---------------------------");

System.out.println("|1 - Книга |");

System.out.println("|2 - Стихи |");

System.out.println("|3 - Комиксы |");

System.out.println("|4 - Выход в главное меню |");

System.out.println("---------------------------");

try {

s = Integer.parseInt(in.nextLine());

if (s < 1 || s > 4) throw new IllegalArgumentException();

} catch (IllegalArgumentException e){

System.out.println("Ошибка ввода, вы можете ввести только 1, 2, 3 или 4");

}

String genre, name, nameBook, nameArt;

switch (s){

case 1:

String a = null;

System.out.print("Имя автора: ");

name = in.nextLine();

System.out.print("Жанр: ");

genre = in.nextLine();

System.out.print("Название книги: ");

nameBook = in.nextLine();

Book book = new Book(genre, name, nameBook);

library.addRuc(book);

break;

case 2:

System.out.print("Имя автора: ");

name = in.nextLine();

System.out.print("Жанр: ");

genre = in.nextLine();

System.out.print("Название стиха: ");

nameBook = in.nextLine();

Poems poems = new Poems(genre, name, nameBook);

library.addRuc(poems);

break;

case 3:

System.out.print("Имя автора: ");

name = in.nextLine();

System.out.print("Имя художника: ");

nameArt = in.nextLine();

System.out.print("Жанр: ");

genre = in.nextLine();

System.out.print("Название книги: ");

nameBook = in.nextLine();

Comix comix = new Comix(genre, name, nameBook, nameArt);

library.addRuc(comix);

break;

}

}while (s != 4);

}

public static void search(){

int b = 0;

do {

System.out.println(" Поиск ");

System.out.println("-------------------------------------");

System.out.println("|1 - По жанрам |");

System.out.println("|2 - Получение всех книг |");

System.out.println("|3 - Получение всех стихов |");

System.out.println("|4 - Получение всех комиксов |");

System.out.println("|5 - Поиск по названию произведения |");

System.out.println("|6 - Выход в главное меню |");

System.out.println("-------------------------------------");

try {

b = Integer.parseInt(in.nextLine());

if (b < 1 || b > 6) throw new IllegalArgumentException();

} catch (IllegalArgumentException e) {

System.out.println("Ошибка ввода, вы можете ввести только 1, 2, 3, 4, 5 или 6");

}

switch (b) {

case 1:

System.out.print("Введите жанр:");

String h = in.nextLine();

ArrayList<Ruc> finded = library.search(h);

if (finded.isEmpty()) {

System.out.println("Рукописей с данным жанром не найдено");

}

else {

for (Ruc a: finded) {

a.look();

}

}

break;

case 2:

System.out.println("Получаем все книги:");

ArrayList<Book> find = library.getAllBooks();

if (find.isEmpty()){

System.out.println("Рукописей с данным жанром не найдено");

} else {

for (Ruc r : find) {

r.look();

}

}

break;

case 3:

System.out.println("Получаем все стихи:");

ArrayList<Poems> finde = library.getAllPoems();

if (finde.isEmpty()){

System.out.println("Рукописей с данным жанром не найдено");

} else {

for (Ruc r : finde) {

r.look();

}

}

break;

case 4:

System.out.println("Получаем все комиксы:");

ArrayList<Comix> fin = library.getAllComix();

if (fin.isEmpty()){

System.out.println("Рукописей с данным жанром не найдено");

} else {

for (Ruc r : fin) {

r.look();

}

}

break;

case 5:

System.out.print("Введите название произведения:");

String q = in.nextLine();

ArrayList<Ruc> f = library.searchNameBook(q);

if (f.isEmpty()){

System.out.println("Рукописей с данным жанром не найдено");

} else {

for (Ruc a : f) {

a.look();

}

}

break;

}

}while (b != 6);

}

public static void delete(){

int n = 0;

String[] rucNames = library.getRucName();

do {

System.out.println("Удаление");

System.out.println("Выбирете какую рукопись удалить");

for (int i = 0; i < rucNames.length; i++) {

System.out.println(i + " )" + rucNames[i]);

}

n = Integer.parseInt(in.nextLine());

}while(n < 0 || n > rucNames.length);

library.delRuc(library.getRucList().get(n));

}

}

2) Ruc

package lab4.data;

public abstract class Ruc {

private String genre;

private String name;

private String nameBook;

public Ruc(String genre, String name, String nameBook){

this.genre = genre;

this.name = name;

this.nameBook = nameBook;

}

public String getGenre(){

return genre;

}

public void setGenre(String genre){

this.genre = genre;

}

public String getName(){

return name;

}

public void setName(String name){

this.name = name;

}

public String getNameBook(){

return nameBook;

}

public void setNameBook(String nameBook){

this.nameBook = nameBook;

}

public abstract void look();

}

3) Book

package lab4.data;

public class Book extends Ruc {

public Book(String genre, String name, String nameBook) {

super(genre, name, nameBook);

}

@Override

public void look() {

System.out.println();

System.out.println("Имя автора: " + getName());

System.out.println("Название: " + getNameBook());

System.out.println("Жанр: " + getGenre());

System.out.println();

}

}

4) Poems

package lab4.data;

public class Poems extends Ruc {

public Poems(String genre, String name, String nameBook) {

super(genre, name, nameBook);

}

@Override

public void look() {

System.out.println();

System.out.println("Имя автора: " + getName());

System.out.println("Название: " + getNameBook());

System.out.println("Жанр: " + getGenre());

System.out.println();

}

}

5) Comix

package lab4.data;

public class Comix extends Ruc {

private String nameArt;

public Comix(String nameArt, String genre, String name, String nameBook) {

super(genre, name, nameBook);

this.nameArt = nameArt;

}

public String getNameArt() {

return nameArt;

}

public void setNameArt (String nameArt) {

this.nameArt = nameArt;

}

@Override

public void look() {

System.out.println();

System.out.println("Имя автора: " + getName());

System.out.println("Имя художника: " + getNameArt());

System.out.println("Название: " + getNameBook());

System.out.println("Жанр: " + getGenre());

System.out.println();

}

}

6)Library

package lab4;

import lab4.data.Book;

import lab4.data.Comix;

import lab4.data.Poems;

import lab4.data.Ruc;

import java.util.ArrayList;

public class Library {

ArrayList<Ruc> rucList = new ArrayList<>();

//добавление рукописей в массив

public void addRuc(Ruc ruc){

rucList.add(ruc);

}

//удаление рукописи из массива

public void delRuc(Ruc ruc){

rucList.remove(ruc);

}

//вывод рукописей

public ArrayList<Ruc> getRucList(){

return rucList;

}

//вывод имен

public String[] getRucName(){

String[] names = new String[rucList.size()];

for (int i = 0; i < rucList.size(); i++){

names[i] = rucList.get(i).getName();

}

return names;

}

//поиск по жанрам

public ArrayList<Ruc> search(String genre){

ArrayList<Ruc> answer = new ArrayList<>();

for (Ruc a:rucList) {

if (a.getGenre().contains(genre)) {

answer.add(a);

}

}

return answer;

}

//получение всех книг

public ArrayList<Book> getAllBooks(){

ArrayList<Book> answer = new ArrayList<>();

for (Ruc a:rucList) {

if (a instanceof Book) {

answer.add((Book) a);

}

}

return answer;

}

//получение всех комиксов

public ArrayList<Comix> getAllComix(){

ArrayList<Comix> answer = new ArrayList<>();

for (Ruc a:rucList) {

if (a instanceof Comix) {

answer.add((Comix) a);

}

}

return answer;

}

//получение всех стихов

public ArrayList<Poems> getAllPoems(){

ArrayList<Poems> answer = new ArrayList<>();

for (Ruc a:rucList) {

if (a instanceof Poems) {

answer.add((Poems) a);

}

}

return answer;

}

//поиск по названию

public ArrayList<Ruc> searchNameBook(String nameBook) {

ArrayList<Ruc> answer = new ArrayList<>();

for (Ruc a : rucList) {

if (a.getNameBook().contains(nameBook)) {

answer.add(a);

}

}

return answer;

}

//вывод информации

public void lookAll(){

for (Ruc a:rucList) {

a.look();

}

}

}

Список использованных источников

1. Методические указания по выполнению лабораторных работ / сост.: В.Л. Аршинский. – Иркутск : Изд-во ИРНИТУ, 2017. – 24 c.
2. Файн Я. – Программирование на Java для детей, родителей, дедушек и бабушек, 2011