Отчет по КСР

Задание №1

Автоматическая генерация тестов на основе формального описания.

Тесты составляются на основе спецификации требований. При формулировании требований на естественном языке существует проблема их различных толкований. Одним из способов избежать этого является применение формальных языков для описания структуры и поведения системы (*UML*, SDL, *MSC*). Кроме того, описание требований на формальном языке является формальным описанием тестовых случаев, на основе которого можно генерировать тестовый код. В практикуме для создания тестов будет использоваться язык диаграмм взаимодействия (Message *Sequence* Charts, *MSC* - п.11). В этом случае под тестом мы будем понимать его *представление* в виде *MSC*- диаграммы.

В Практикуме для реализации тестирования используется учебная система *автоматизации тестирования* *TAT* - *Test Automation* *Training*. На вход система принимает формальное описание тестов в виде *MSC* диаграмм (в текстовом формате *MSC* PR). На основе этих *MSC* диаграмм и конфигурационного файла (в формате *XML*), который описывает *интерфейс* тестируемой системы, генерируется тест на С#. (*Интерфейс* тестируемого приложения (*Application* Under Test - AUT) содержит сигналы, сообщения, транзакции, которые система может посылать тестовому окружению или может принимать от тестового окружения). Для запуска системы с этим тестом необходимо написать *Wrapper*, который транслирует сигналы от теста к системе и наоборот.

Таким образом, методика тестирования системы с помощью *TAT* выглядит следующим образом:

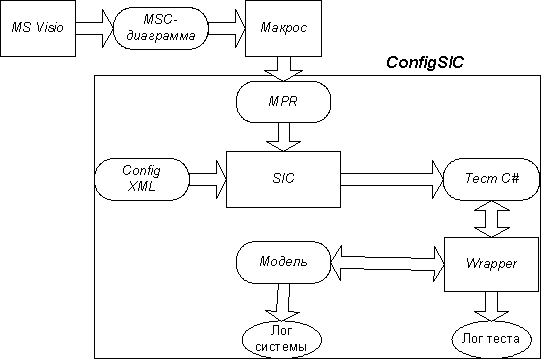
* написать *Wrapper* для тестируемой системы;
* создать файл конфигурации;
* создать формальное описание тестов в виде MSC-диаграмм;
* нарисовать MSC-диаграммы в MS Visio;
* сгенерировать с помощью макроса тестовый файл в формате *MPR*;
* настроить в ConfigTAT проект теста (указать пути) или набора тестов;
* запустить тест или набор тестов;
* проанализировать получаемые *log-файлы*.

В данном случае *wrapper* и *файл* конфигурации (первые два пункта методики) уже созданы, поэтому вам необходимо будет выполнить только п. 3-6.

В рассматриваемом подходе не только *запуск* тестов и проверка результатов прогона **тестового случая** будут осуществляться автоматически, но и сам тестовый код будет генерироваться автоматически на основе *MSC*-диаграммы (рис. 1).

При разработке тестов был использован следующий подход:

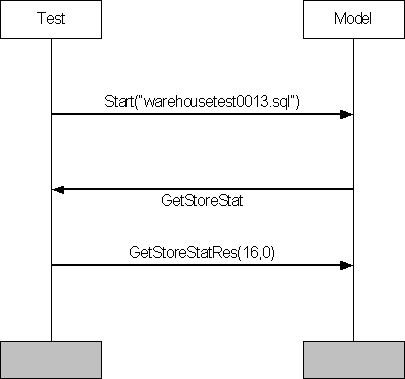
Когда реализуется определенное событие, модель посылает сигнал запроса состояния к тестовому окружению.



**Рис. 1.**Система и ее окружение (автоматическая генерация)

Состояние окружения задается в тесте (**входные данные**) в виде параметров сигналов. Тест возвращает состояние окружения (StoreStat, AxlePar, RollerPar, StoreMessage, CommandStatus), посылая модели сигнал с параметрами в соответствии с запросом.

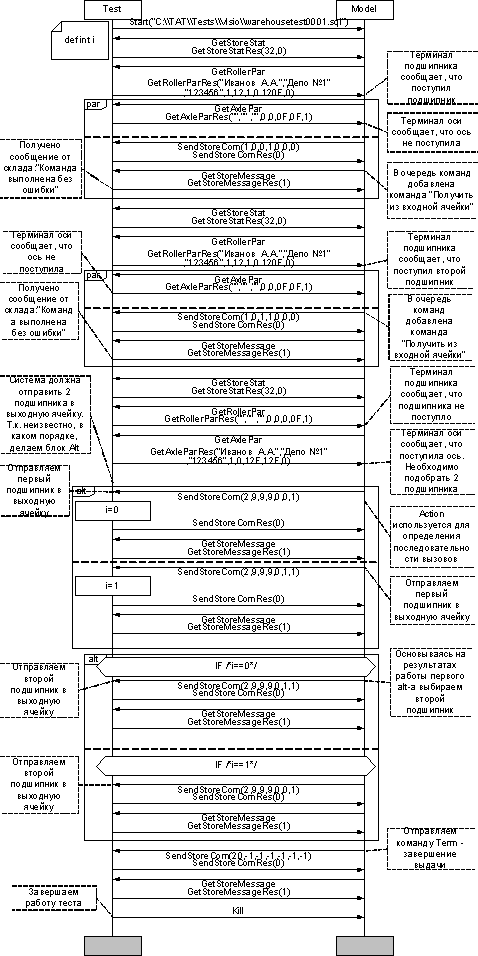
Получаемая *информация* сохраняется в журнале теста.



**Рис. 2.**Взаимодействие теста и модели (MSC-диаграмма)

**Подробное описание тестового случая №1**

Изучение материала настоящего пункта полезно предварить ознакомлением с п. 9, содержащим описание подхода к автоматической генерации тестов на основе *MSC*. Здесь рассматривается тест, представляющий собой *MSC*-диаграмму, созданную в Visio. Подобные тесты необходимо будет разработать (нарисовать) самостоятельно при выполнении заданий. Приведенный пример был разработан в соответствии со спецификацией тестового случая №1 (рис. 3).



**Рис. 3.**Тестовый случай №1

### Описание тестовых процедур

#### Как сгенерировать и запустить тест

Изучение материала настоящего пункта полезно предварить ознакомлением с п. 12, 13, содержащими описание использования MS Visio для генерации *MPR* файлов и описание конфигурирования - ConfigTAT.

На данном этапе используется тест, представляющий собой MSC- диаграмму, созданную в Visio. Дальнейшие действия описываются следующей методикой:

1. Запустить Microsoft Visio.
2. Загрузить Stensil ( File->Open->MSC.*VSS* или File->Open Stensil ->MSC.*VSS* ). Visio выдаст предупреждение о том, что данный stensil содержит макросы. На предупреждение следует ответить "Enable macros".
3. Открыть существующий тестовый случай №1 - Warehousetest0001 (..\SystemTesting\TATTests\Tests\Tests\Тests.*vsd*).
4. Для генерации *MPR* вызвать следующий макрос: Tools->Macros ->MSC->Module1->Parse. В указанной папке будет создан *MPR*-файл с именем, соответствующим имени текущей страницы в Visio (..\SystemTesting\TATTests\Tests\WarehouseTest1\warehousetest0001.*mpr*).
5. Запустить ConfigTAT.
6. В меню File -> Open выбрать тестовый случай №1.
7. Выбрать настройки - установить по умолчанию (Set ALL to default).
8. Запуск - генерация и запуск теста (Run ALL).

#### Проверка результатов выполнения тестов (сравнение с ожидаемым результатом)

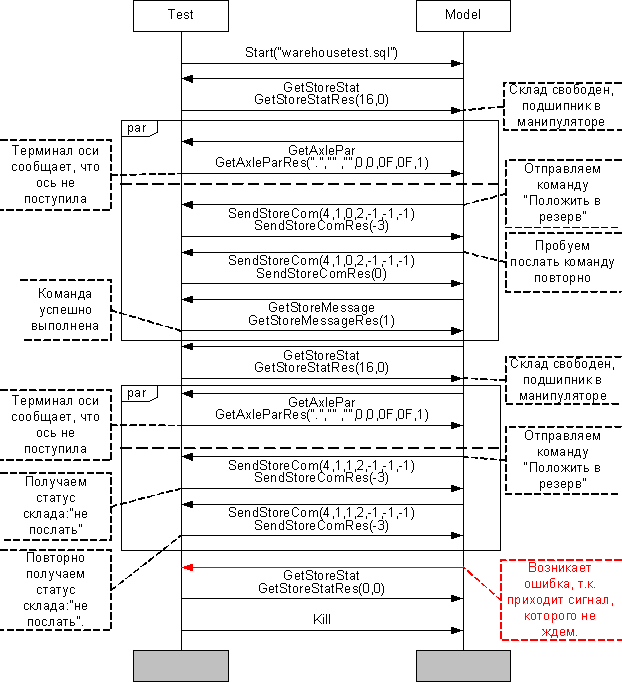
В этом случае запуск тестов и проверка того, что тестируемая система прошла испытания на заданном **тестовом случае**, осуществляется автоматически, как и в предыдущем случае создается журнал теста, а также можно использовать SystemLogAnimator (п.14) для визуализации журнала системы.

Для просмотра протоколов тестирования надо использовать группу "*Test Logs*" ConfigTAT и можно просматривать:

* протокол тестирования в виде html-страницы (HTML log).
* протокол тестирования в виде txt файла (Text-log).
* протоколы в формате *mpr* (отдельный протокол для каждого testcase- а и каждой итерации теста), которые можно открыть в программе Telelogic нажатием кнопки "View" (*MPR* logs).

### Пример теста с ошибкой

На рис. 4 представлена *диаграмма* теста с ошибкой. Используя FS, необходимо объяснить причину некорректности тестового случая.



**Рис. 4.**Тест с ошибкой

Задание №2

Были даны 3 двумерных массива, размерности 3 на 3. Нужно было заполнить первый массив случайными числами и вывести его на экран. Заполнить второй массив с клавиатуры и вывести его на экран. Создать третий массив сложением 1-го и 2-го массива и вывести его на экран. Найти и вывести на экран максимальный элемент 3-го массива и среднеарифметическое элементов главной и побочной диагонали. Рис. 5 – код программы, рис. 6 – продолжение кода, рис. 7 - вывод в консоль.

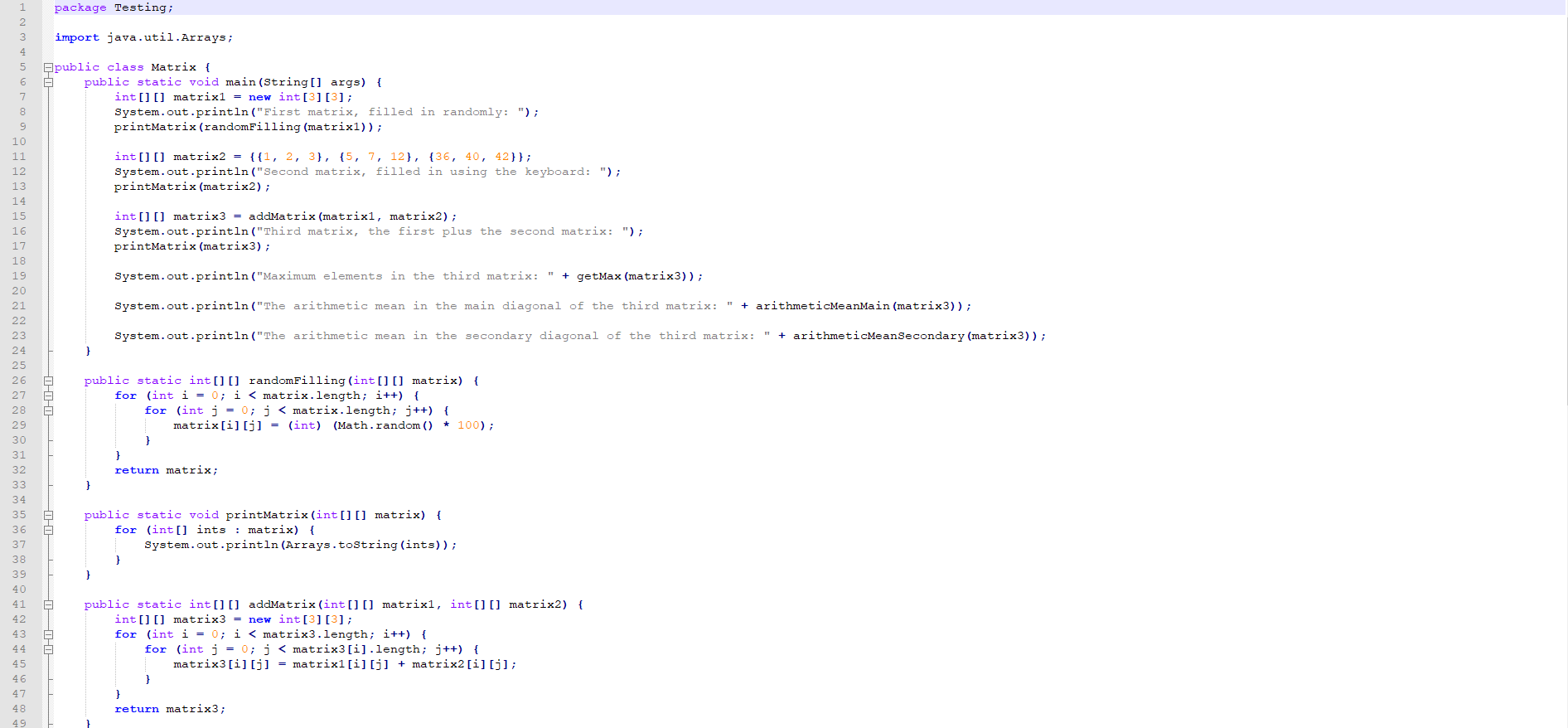


Рис. 5



Рис. 6

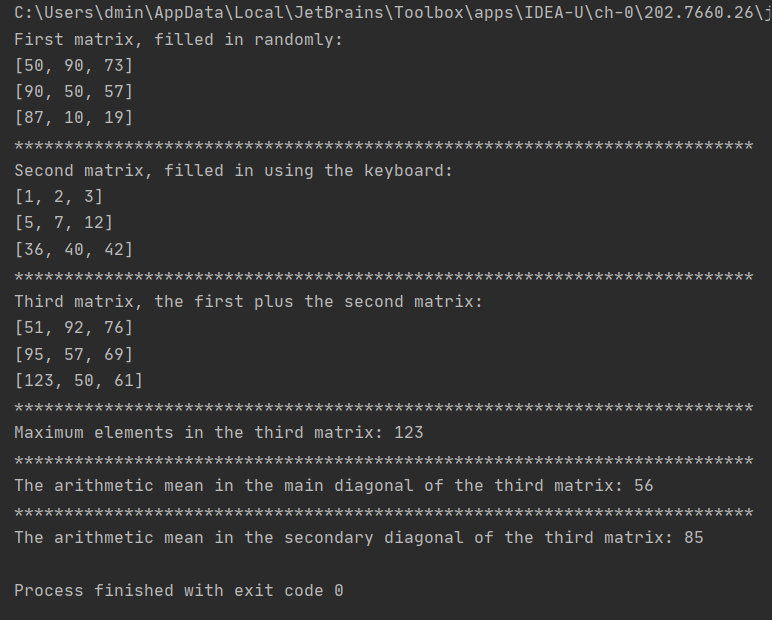


Рис. 7

Отчет по дополнительным задачам

Задание №1

Нужно было написать программу проверки, является ли введенная с клавиатуры квадратная матрица «магическим квадратом» Результат вывести на экран. При решении использовать функции. Рис. 8 – код программы, рис.9 – продолжение кода, рис. 10 – вывод в консоль.

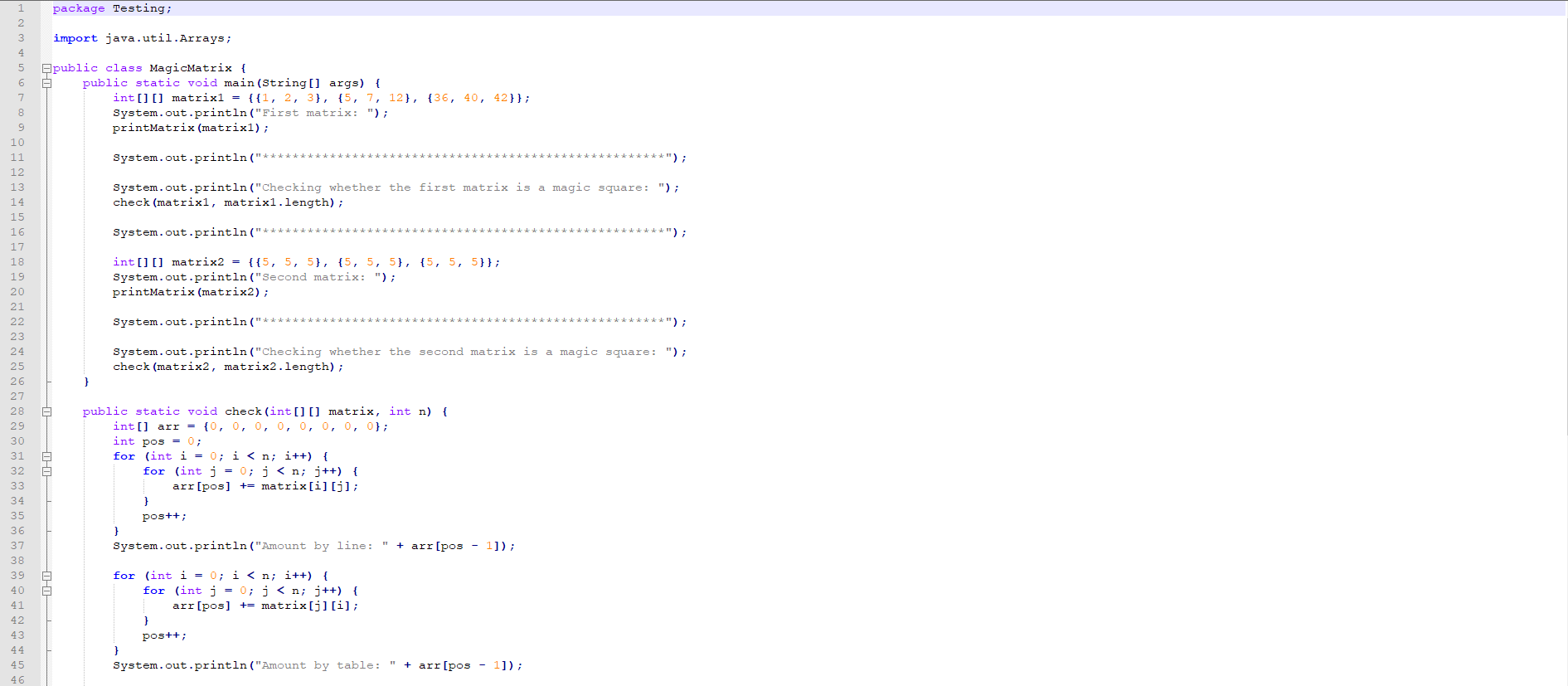
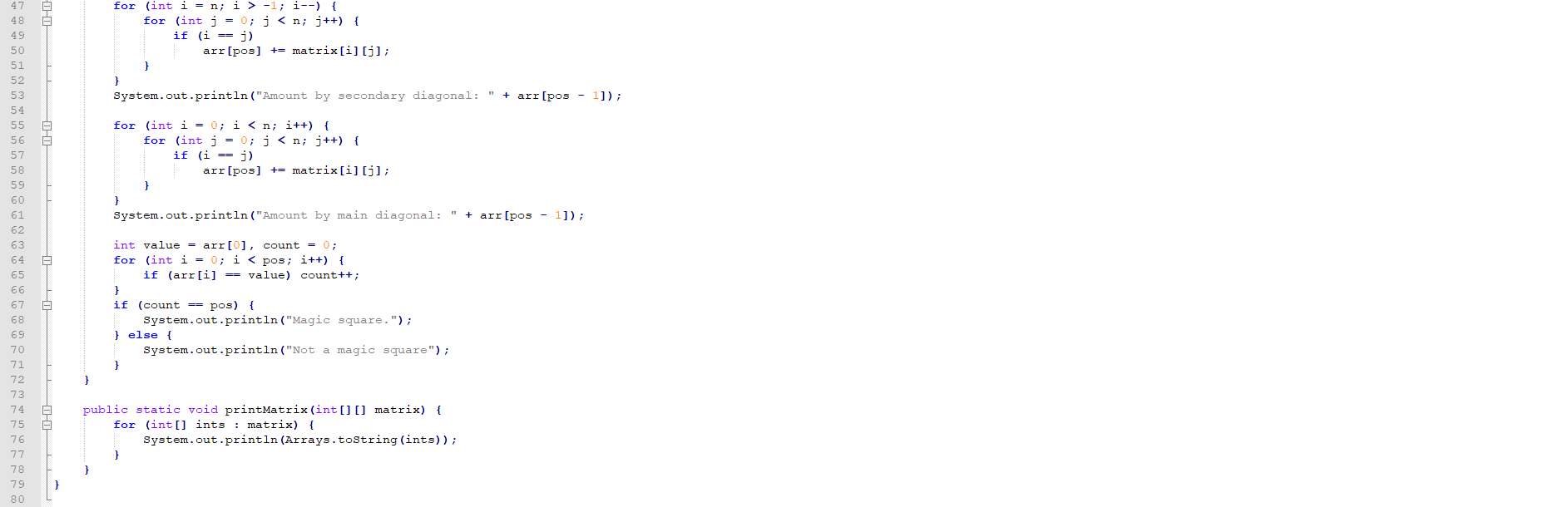


Рис. 8



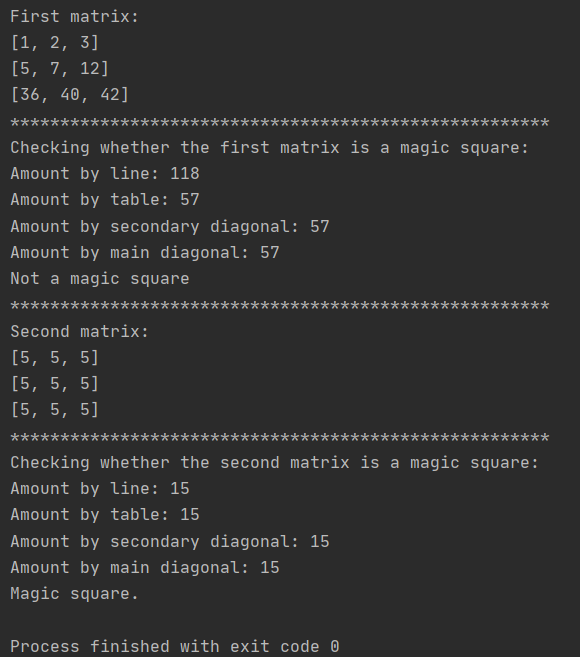
Рис. 9

Рис. 10

Задание №2

Нужно было написать программу проверки, находится ли введенное с клавиатуры число в массиве. Массив заполняется пользователем с клавиатуры. Результат вывести на экран. Рис. 11 – код программы, рис. 12 – вывод в консоль.

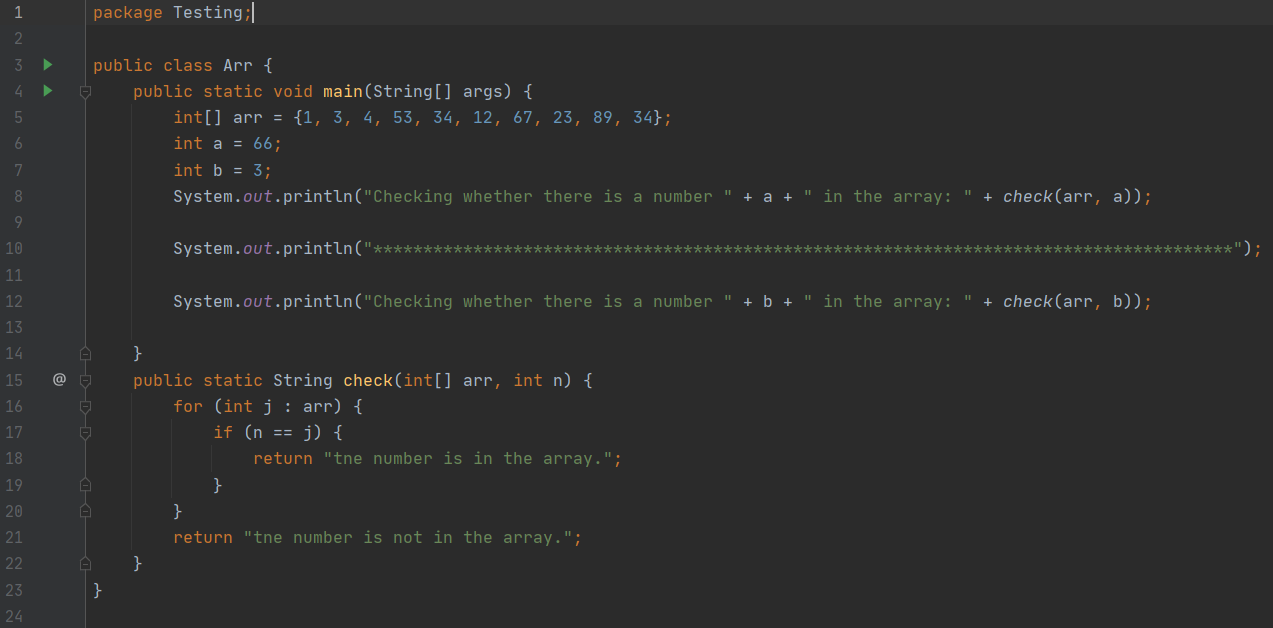


Рис. 11

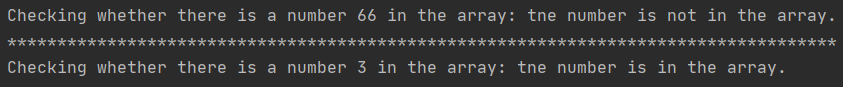


Рис. 12

Задание №4

Нужно было написать программу нахождения суммы четных, нечетных и нулевых элементов массива (используя функции). Массив заполняется пользователем с клавиатуры. Результат вывести на экран. Рис. 13 – код программы, рис. 14 – продолжение, рис. 15 – вывод в консоль.

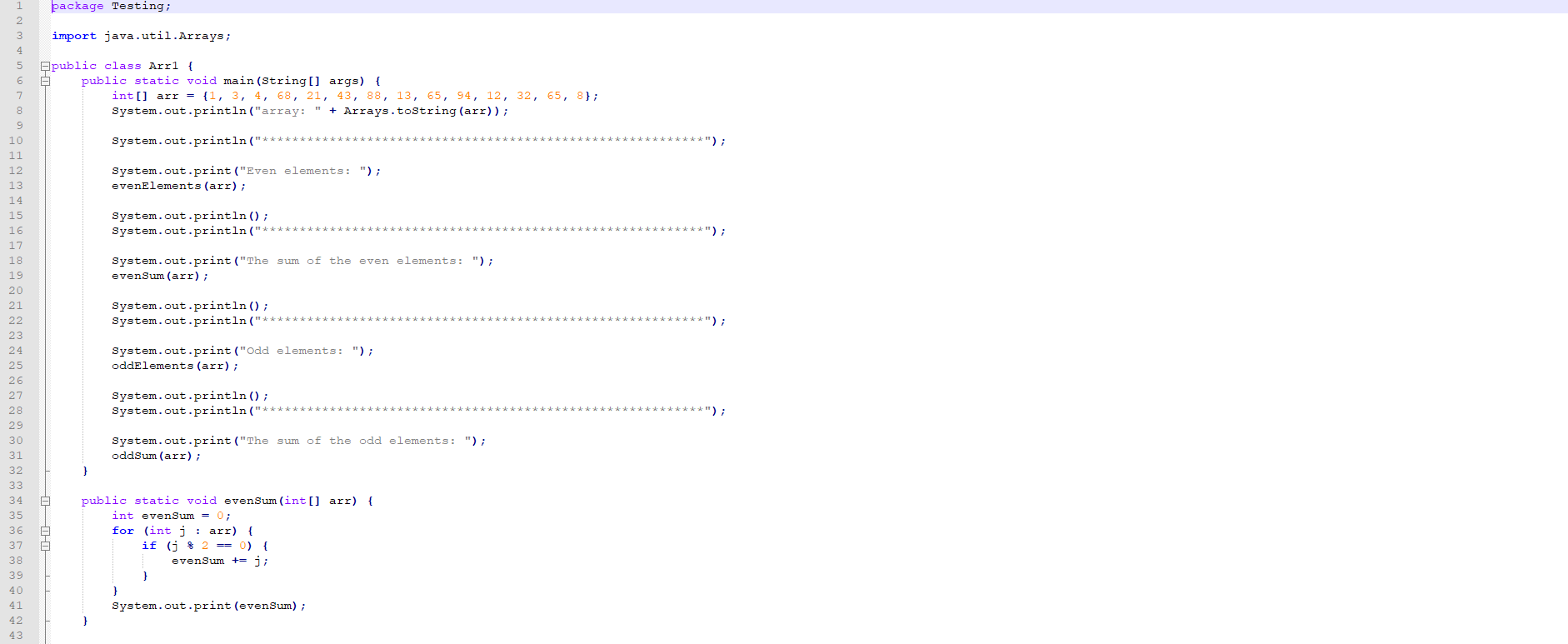


Рис. 13



Рис. 14

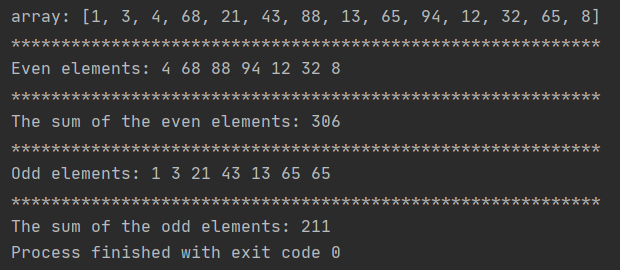


Рис.15