

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования

Петрозаводский государственный университет

Институт математики и информационных технологий

Кафедра информатики и математического обеспечения

Отчет по дисциплине

«Верификация ПО»

ПРОГРАММНАЯ СИСТЕМА РАСПОЗНАВАНИЯ ЛИЦ НА
ИЗОБРАЖЕНИЯХ С НИЗКИМ КАЧЕСТВОМ СЪЕМКИ

Выполнил:

магистрант 2 года группы 22608 Г. Э. Рого

Лектор: к.ф-м.н., доцент К. А. Кулаков

Итоговая оценка: _____

Петрозаводск

2018

Оглавление

1. Описание объекта тестирования.....	3
2. Архитектура программы.....	4
3. Используемые материалы для тестирования	5
4. План тестирования.....	6
5. Журнал тестирования.....	14
6.Отчеты об ошибках.....	23
7. Метод покрытия.....	23
8. Результаты.....	23

Описание объекта тестирования.

Объектом тестирования является консольное приложение, позволяющее распознавать лица людей на фотографиях с низким качеством съемки. Приложение написано на языке Python с использованием библиотеки OpenCV. Приложение реализовано для использования в операционной системе Windows.

Основные функции.

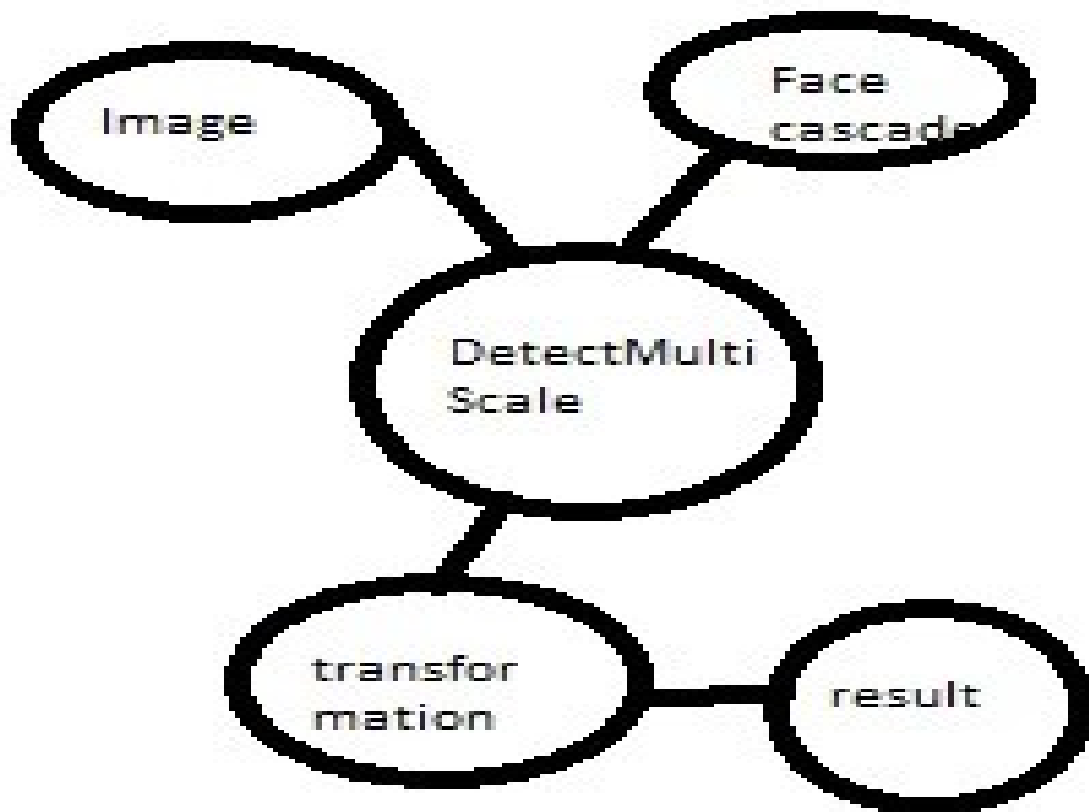
Основные функции объекта тестирования, предназначенные для использования конечным пользователем:

1. Загрузка изображения, на котором необходимо распознать лица
2. Выставление параметров поиска (полнота и точность)

Программа работает с изображениями низкого качества. Таковыми считаются изображения, сделанные ранее 2000-го года, либо засвеченные естественным освещением (солнце). Более формальным критерием является низкий процент распознавания лиц на изображениях при запуске стандартных функций распознавания с рекомендованными параметрами.

Архитектура программы.

Архитектура программы состоит из модулей библиотеки OpenCV и стандартных методов языка Python.



Центральным модулем программы является модуль DetectMultiScale, который и осуществляет распознавание лиц.

Модуль Image отвечает за загрузку и предварительную обработку изображения.

Модуль Face cascade загружает натренированную модель, на которую DetectMultiScale опирается при распознавании.

Модуль Transformation производит пост-обработку изображения и его подготовку к выводу пользователю.

Модуль Result выводит распознанное изображение на экран пользователя.

Используемые материалы для тестирования.

Для тестирования используются изображения, с которыми не справились стандартные функции распознавания, запущенные с рекомендованными параметрами. Количество пикселей в данном контексте значения не имеет, потому что все изображения были оцифрованы на высококачественных сканерах, поэтому данный параметр не отражает реального качества фотографий.

По заданному параметру были отобраны 10 фотографий, на которых и производилось тестирование программы. Полнота при распознавании лиц стандартными методами с рекомендуемыми параметрами составила <50%, а точность <20%. Изображения были пронумерованы от 11 до 20. На каждом фото от 10 до 15 лиц. Лицо засчитывается за такое, если угол поворота по отношению к фотографу не >90%.

Пример изображения (18.jpg):



Кроме того, используются изображения 11.jpg-15.jpg, растянутые в ширину или высоту, обозначенные 21.jpg...25.jpg.

План тестирования.

Процедура тестирования

Основные положения процедуры проведения тестирования:

1. В рамках курса «Верификация ПО», тестирование будет проведено один раз, то есть без повторного проведения и исправления найденных ошибок;
2. Процедура тестирования будет остановлена, только в случае нахождения блокирующих ошибок;
3. Тестирование будет проводиться автором данного отчета, без привлечения других лиц;
4. Процедура тестирования будет признана выполненной успешно, если в ходе проведения не будет найдено блокирующих или критических ошибок.

Стратегия блочного тестирования.

Первый вид тестирования, которому будет подвержена система - блочное тестирование. Его цель: проверить работу функций в зависимости от различных входных параметров. Данный вид тестирования будет применен **ко всем функциям** программы. Так как весь исходный код модулей написан на языке Python, то для блочного и интеграционного тестирования будет использоваться инструмент PyUnit там, где это возможно и целесообразно.

Автоматические тесты с помощью PyUnit включают в себя определение действий, производимых до теста, определение входных данных, тестирование функций с помощью утверждений (проверка выходных значений или выброс исключений), действия после теста.

Список функций для тестирования:

- CascadeClassifier
- imread
- resize

- INTER_AREA
- cvtColor
- detectMultiScale
- rectangle
- namedWindow
- imshow

Стратегия интеграционного тестирования

Второй этап - интеграционное тестирование. Для проведения интеграционного тестирования необходимо определить все возможные входные данные, соответствующие им ожидаемые результаты.

Будет проверяться интеграция модуля detectmultiscale и rectangle, протестированных блочно, на предыдущем этапе. Интегрироваться модуль будет по принципу использования функций из модуля detectmultiscale через вызов ядром rectangle.

Схематично порядок прохождения теста можно изобразить сл. схемой:



Аттестационное тестирование

Аттестационное тестирование будет проводится методом «Живого человека». В роли такого человека выступает сам автор тестирования.

Тестирующий человек, по заранее заданным инструкциям (Test Cases), производит требуемые действия и сверяется с заранее заданными результатами. Тест считается пройденным, если в результате полученные такие же данные, которые описаны в инструкции. В противном случае тест считается негативным.

Для проведения тестирования в данном случае необходим терминал и интерпретатор Python 3.

Аттестация системы будет производиться по следующим высокоуровневым функциям:

- 1) Загрузка изображения и выставление параметров поиска лиц
- 2) Подача на вход программе входных изображений разных размеров

Специальное тестирование.

Проверка масштабирования.

На вход программе подаются изображения разного качества, происходит проверка способности программы работать с изображениями различного качества.

Критерием качества фотографий является способность стандартных алгоритмов распознать лица на изображении. Если лица распознаются с высокой точностью (более 80%), то изображение является качественным, иначе некачественным.

Критерий прохождения тестов

Тест считается успешно пройденным, если ожидаемый и фактический результаты совпадают. Если тест завершается неудачей, то перед принятием решения целесообразно проверить правильность самого теста. Если тест завершился неудачей и тест реализован правильно, то производится заключение о найденной ошибке.

Тестирование считается пройденным, если во время его прохождения не выявлено критических ошибок, а процент не пройденных тестов меньше 1% от общего количества.

Критерий приостановки тестов

Тестирование должно быть приостановлено, если количество не пройденных тестов превысит 10% от их общего количества. Тестирование должно быть приостановлено при обнаружении критических ошибок.

Критерий возобновления тестирования

Тестирование возобновляется после исправления ошибок, выявленных при предыдущем тестировании, при условии, что найденные ошибки являются критическими и без их исправления нельзя продолжать тестирование.

Описание тестов

- П – позитивный
- С – специальный

- Н – негативный

Блочные тесты

Тест	1-П
Описание	Получение координат всех лиц на изображении
Функция	detectMultiScale(image[, scaleFactor[, minNeighbors[, flags[, minSize[, maxSize]]]])
Входные параметры	[gray, 1.1, 1]
Ожидаемый результат	Список координат найденных лиц
Тест	1-Н
Описание	Получение координат всех лиц на изображении
Функция	detectMultiScale(image[, scaleFactor[, minNeighbors[, flags[, minSize[, maxSize]]]])
Входные параметры	[gray, 1.0001, 1]
Ожидаемый результат	Ошибка памяти
Тест	2-П
Описание	Загрузка модели распознавания лиц
Функция	CascadeClassifier
Входные параметры	[haarcascade_frontalface_default.xml]
Ожидаемый результат	Продолжение работы программы
Тест	2-Н
Описание	Загрузка некорректной модели распознавания лиц
Функция	CascadeClassifier

Входные параметры	[haarcascade_frontalface_default2.xml]
Ожидаемый результат	Ошибка о некорректном содержании файла
Тест	3-Н
Описание	Загрузка некорректного файла
Функция	Imread
Входные параметры	[1.txt]
Ожидаемый результат	Ошибка о некорректном формате файла
Тест	4-П
Описание	Изменение размеров изображения
Функция	Resize
Входные параметры	10, 100, 1000, 10000
Ожидаемый результат	Изображение с новыми размерами
Тест	5-П
Описание	Интерполяция изображения
Функция	INTER_AREA
Входные параметры	[]
Ожидаемый результат	Интерполированное изображение
Тест	6-П
Описание	Преобразование изображения в черно-белый формат
Функция	cvtColor
Входные параметры	[resized (изображение с измененными размерами) cv2.COLOR_BGR2GRAY (функция преобразования цвета)]
Ожидаемый результат	gray (Изображение в серых тонах)
Тест	7-П
Описание	Отрисовка изображений полученных лиц
Функция	Rectangle
Входные параметры	Список координат лиц faces = [[699, 382, 57, 57], [268, 572, 60, 60], [629, 234, 50, 50], [278, 256, 57, 57], [673, 272, 25, 25], [532, 58, 77, 77], [704, 259, 56, 56], [171, 269, 72, 72]]
Ожидаемый результат	Изображение
Тест	8-П
Описание	Получение количества всех принадлежащих городу

	апартаментов в виде массива объектов
Функция	namedWindow
Входные параметры	['resized', imshow]
Ожидаемый результат	Созданное окно с заданным названием
Тест	9-П
Описание	Получение количества всех принадлежащих городу апартаментов в виде массива объектов
Функция	namedWindow
Входные параметры	['123e4', imshow]
Ожидаемый результат	Созданное окно с заданным названием
Тест	4-Н
Описание	Получение количества всех принадлежащих городу апартаментов в виде массива объектов
Функция	namedWindow
Входные параметры	['dsfjh123N%;%:?', imshow]
Ожидаемый результат	Созданное окно с заданным названием
Тест	10-П
Описание	Выведение окна в интерфейс пользователя
Функция	imshow
Входные параметры	['resized', resized]
Ожидаемый результат	Вывод окна

Интеграционные тесты

Тест	11-П
Описание	Получение списка координат найденных лиц и их прорисовка на изображении
Функция	detectmultiscale
Взаимодействие	Rectangle
Входные параметры	[11.jpg-20.jpg]
Ожидаемый результат	(bool) true
Тест	5-Н
Описание	Проверка реакции на некорректное взаимодействие
Функция	Detectmultiscale
Взаимодействие	Rectangle
Входные параметры	Результат работы detectmultiscale +(['saffd', 1234,543,234], [+ ,1234,543,234], [1234, 12.3, 543, 234],
Ожидаемый результат	Ошибки работы функции Rectangle

Аттестационные тесты

Тест	12-П
Описание	Распознавание лиц на изображении с рекомендуемыми параметрами
Входные параметры	(11.jpg...20.jpg),(1.1-шаг окна сканирования, 1-количество соседей, по которым идет классификации)
Ожидаемый результат	Изображение с распознанными лицами
Тест	13-П
Описание	Быстрое распознавание лиц на изображении
Входные параметры	(11.jpg...20.jpg),(1.9-шаг окна сканирования, 3-количество соседей, по которым идет классификации)
Ожидаемый результат	Программа должна завершить распознавание в течение 10 секунд
Тест	14-П
Описание	Распознавание лиц на изображениях нестандартных размеров
Входные параметры	(21.jpg...25.jpg-изображения, растянутые в высоту или ширину), (1.1-шаг окна сканирования, 1-количество соседей, по которым идет классификации)
Ожидаемый результат	Изображения с распознанными лицами

Специальные тесты

Тест	1-С
Описание	Загрузка изображений различных расширений
Функция	Imread
Входные параметры	[photo ***], где *** - 10, 100, 1000, 10000
Ожидаемый результат	Корректная загрузка изображения
Тест	2-С
Описание	Распознавание лиц на изображении с экстремальными параметрами
Входные параметры	(11.jpg...20.jpg),(1.001-шаг окна сканирования, 15-количество соседей, по которым идет классификации)
Ожидаемый результат	Изображение с распознанными лицами
Тест	3-С
Описание	Проверка работы алгоритма на качественных изображениях
Функция	Программа целиком
Входные параметры	(1.jpg...10.jpg),(1.01-шаг окна сканирования, 1-количество соседей, по которым идет классификации)
Ожидаемый результат	Распознанные лица на изображениях

Журнал тестирования

Все тесты проведены 01.12.2018.

Блочные тесты

Тест	1-П
Описание	Получение координат всех лиц на изображении
Функция	<code>detectMultiScale(image[, scaleFactor[, minNeighbors[, flags[, minSize[, maxSize]]]])</code>
Входные параметры	<code>[gray, 1.1, 1]</code>
Ожидаемый результат	Список координат найденных лиц
Результат	<pre>faces = [[55, 157, 160, 160], [699, 382, 57, 57], [268, 572, 60, 60], [239, 221, 25, 25], [252, 331, 189, 189], [827, 638, 25, 25], [624, 406, 25, 25], [13, 595, 80, 80], [498, 657, 30, 30], [510, 254, 33, 33], [832, 426, 80, 80], [629, 234, 50, 50], [278, 256, 57, 57], [154, 186, 212, 212], [673, 272, 25, 25], [910, 435, 53, 53], [532, 58, 77, 77], [704, 259, 56, 56], [171, 269, 72, 72], [698, 285, 59, 59], [455, 325, 25, 25], [794, 506, 28, 28], [521, 354, 26, 26], [759, 337, 34, 34]]</pre>

Тест	1-Н
Описание	Получение координат всех лиц на изображении
Функция	detectMultiScale(image[, scaleFactor[, minNeighbors[, flags[, minSize[, maxSize]]]])
Входные параметры	[gray, 1.0001, 1]
Ожидаемый результат	Ошибка памяти
Результат	cv2.error: C:\projects\opencv-python\opencv\modules\core\src\matrix.cpp:436: error: (-215) u != 0 in function cv::Mat::create
Тест	2-П
Описание	Загрузка модели распознавания лиц
Функция	CascadeClassifier
Входные параметры	[haarcascade_frontalface_default.xml]
Ожидаемый результат	Продолжение работы программы
Результат	Программа продолжила работу
Тест	2-Н
Описание	Загрузка некорректной модели распознавания лиц
Функция	CascadeClassifier
Входные параметры	[haarcascade_frontalface_default2.xml]
Ожидаемый результат	Ошибка о некорректном содержании файла
Результат	File: haarcascade_frontalface_default2.xml is incorrect
Тест	1-С
Описание	Загрузка изображений различных расширений
Функция	Imread
Входные параметры	[photo ***], где *** - 10, 100, 1000, 10000
Ожидаемый результат	Корректная загрузка изображения
Результат	Запусков: 4 Ошибок: 1 Исправлений: 0 Успешность: 25%
Тест	3-Н
Описание	Загрузка некорректного файла
Функция	Imread
Входные параметры	[1.txt]
Ожидаемый результат	Ошибка о некорректном формате файла
Результат	File: "1.txt" is not image
Тест	4-П
Описание	Изменение размеров изображения
Функция	Resize
Входные параметры	10, 100, 1000, 10000
Ожидаемый результат	Изображение с новыми размерами

Результат	Запусков: 4 Ошибок: 0 Исправлений: 0 Успешность: 100%
Тест	5-П
Описание	Интерполяция изображения
Функция	INTER_AREA
Входные параметры	[]
Ожидаемый результат	Интерполированное изображение
Результат	Интерполированное изображение
Тест	6-П
Описание	Преобразование изображения в черно-белый формат
Функция	cvtColor
Входные параметры	[resized (изображение с измененными размерами) cv2.COLOR_BGR2GRAY (функция преобразования цвета)]
Ожидаемый результат	gray (Изображение в серых тонах)
Результат	gray (Изображение в серых тонах)
Тест	7-П
Описание	Отрисовка изображений полученных лиц
Функция	Rectangle
Входные параметры	Список координат лиц faces = [[699, 382, 57, 57], [268, 572, 60, 60], [629, 234, 50, 50], [278, 256, 57, 57], [673, 272, 25, 25], [532, 58, 77, 77], [704, 259, 56, 56], [171, 269, 72, 72]]
Ожидаемый результат	Изображение
Результат	Запусков: 1 Итераций: 8 Ошибок: 0 Исправлений: 0 Успешность: 100%
Тест	8-П-П, 4-Н

Описание	Получение количества всех принадлежащих городу апартаментов в виде массива объектов
Функция	namedWindow
Входные параметры	['dsfjh123N%;%:?', imshow]
Ожидаемый результат	Созданное окно с заданным названием
Результат	Запусков: 1 Итераций: 1 Ошибок: 0 Исправлений: 0 Успешность: 100%
Тест	8-П
Описание	Получение количества всех принадлежащих городу апартаментов в виде массива объектов
Функция	namedWindow
Входные параметры	['resized', imshow]
Ожидаемый результат	Созданное окно с заданным названием
Результат	Запусков: 1 Итераций: 1 Ошибок: 0 Исправлений: 0 Успешность: 100%
Тест	9-П
Описание	Получение количества всех принадлежащих городу апартаментов в виде массива объектов
Функция	namedWindow
Входные параметры	['123e4', imshow]
Ожидаемый результат	Созданное окно с заданным названием
Результат	Запусков: 1 Итераций: 1 Ошибок: 0 Исправлений: 0 Успешность: 100%
Тест	10-П
Описание	Выведение окна в интерфейс пользователя
Функция	imshow
Входные параметры	['resized', resized]
Ожидаемый результат	Вывод окна
Результат	Запусков: 10 Ошибок: 0 Исправлений: 0

	Успешность: 100%
--	------------------

Интеграционные тесты

Тест	11-П
Описание	Получение списка координат найденных лиц и их прорисовка на изображении
Функция	detectmultiscale
Взаимодействие	Rectangle
Входные параметры	[11.jpg-20.jpg]
Ожидаемый результат	(bool) true
Результат	Запусков: 10 Итераций: 1 Ошибок: 0 Исправлений: 0 Успешность: 100%
Тест	5-Н
Описание	Проверка реакции на некорректное взаимодействие
Функция	Detectmultiscale
Взаимодействие	Rectangle
Входные параметры	Результат работы detectmultiscale +(['saffd', 1234, 543, 234], [+ , 1234, 543, 234], [1234, 12.3, 543, 234],
Ожидаемый результат	Ошибки работы функции Rectangle
Результат	Запусков: 3 Итераций: 1 Ошибок: 0 Исправлений: 0 Успешность: 100%

Аттестационные тесты

Тест	12-П
Описание	Распознавание лиц на изображении с рекомендуемыми параметрами
Входные параметры	(11.jpg...20.jpg),(1.1-шаг окна сканирования, 1-количество соседей, по которым идет классификации)
Ожидаемый результат	Изображение с распознанными лицами
Результат	Запусков: 10

	<p>Ошибок: 0</p> <p>Исправлений: 0</p> <p>Успешность: 100%</p>
Тест	2-С
Описание	Распознавание лиц на изображении с экстремальными параметрами
Входные параметры	(11.jpg...20.jpg),(1.001-шаг окна сканирования, 15-количество соседей, по которым идет классификации)
Ожидаемый результат	Изображение с распознанными лицами
Результат	<p>Запусков: 10</p> <p>Ошибок: 7</p> <p>Исправлений: 0</p> <p>Успешность: 30%</p>
Тест	13-П
Описание	Быстрое распознавание лиц на изображении
Входные параметры	(11.jpg...20.jpg),(1.9-шаг окна сканирования, 3-количество соседей, по которым идет классификации)
Ожидаемый результат	Программа должна завершить распознавание в течение 10 секунд
Результат	<p>Запусков: 10</p> <p>Ошибок: 0</p> <p>Исправлений: 0</p> <p>Среднее время: 4.28472 секунд</p> <p>Успешность: 100%</p>
Тест	14-П
Описание	Распознавание лиц на изображениях нестандартных размеров
Входные параметры	(21.jpg...25.jpg-изображения, растянутые в высоту или ширину), (1.1-шаг окна сканирования, 1-количество соседей, по которым идет классификации)
Ожидаемый результат	Изображения с распознанными лицами
Результат	<p>Запусков: 5</p> <p>Ошибок: 0</p> <p>Исправлений: 0</p> <p>Успешность: 100%</p>

Специальное тестирование

Тест	1-С
Описание	Загрузка изображений различных расширений

Функция	Imread
Входные параметры	[photo ***], где *** - 10, 100, 1000, 10000
Ожидаемый результат	Корректная загрузка изображения
Результат	Запусков: 4 Ошибок: 1 Исправлений: 0 Успешность: 25%
Тест	2-С
Описание	Распознавание лиц на изображении с экстремальными параметрами
Входные параметры	(11.jpg...20.jpg),(1.001-шаг окна сканирования, 15-количество соседей, по которым идет классификации)
Ожидаемый результат	Изображение с распознанными лицами
Результат	Запусков: 10 Ошибок: 7 Исправлений: 0 Успешность: 30%
Тест	3-С
Описание	Проверка работы алгоритма на качественных изображениях
Функция	Программа целиком
Входные параметры	(1.jpg...10.jpg),(1.01-шаг окна сканирования, 1-количество соседей, по которым идет классификации)
Ожидаемый результат	Распознанные лица на изображениях
Результат	Запусков: 10 Ошибок: 0 Исправлений: 0 Точность распознавания: 96%

Пример теста.

```
Import cv2
```

```
Import unittest
```

```
From scipy import ndimage
```

```
Import numpy as np
```

```
Data= detectMultiScale([gray, 1.1, 1])
```

Try:

```
Data = Data(int(['*', '*', '*']))
```

Except ValueError:

```
Print ("Неверный формат данных")
```

Отчеты об ошибках

Отчет №1

Краткое описание: Ошибка при выполнении теста 1-С. Загрузка изображения размеров 10000 и 100000 не завершилась успехом.

Ожидаемый результат: Загрузка изображения и возможность работы с ним.

Фактический результат: Программа перестала отвечать на запросы и её пришлось закрывать, используя стандартные методы windows.

Отчет №2

Краткое описание: Ошибка при выполнении теста 2-С. Повышение полноты поиска до 99,999% привело к переполнению памяти и зависанию программы.

Ожидаемый результат: Загрузка изображения с распознанными лицами.

Фактический результат: Произошло переполнение памяти и программа завершила работу.

Методы покрытия

Расчёт покрытия тестами относительно исполняемого кода производится по формуле:

Covering = Testing_lenght/Code_lenght.

Testing_lenght - количество строк кода, покрытых тестами = 352

Code_lenght - общее количество строк кода в программе = 417

Коэффициент покрытия: 0,84

Процент покрытия: 84,4%

Результаты

В ходе выполнения тестов было выявлено несколько ошибок, которые представлены в отчетах тестирования.

По полученным результатам тестирования, можно судить, что объект готов к работе после доработок. Критических ошибок, которые требовалось бы исправлять срочно, не обнаружено.