

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
Петрозаводский государственный университет
Математический факультет
Кафедра информатики и математического обеспечения

Отчет по дисциплине «Верификация программного обеспечения»

ТЕСТИРОВАНИЕ ПРОГРАММЫ РАСЧЕТА
ЖЕЛЕЗОБЕТОННЫХ КОНСТРУКЦИЙ СТЕН МЕТОДОМ
КОНЕЧНЫХ ЭЛЕМЕНТОВ

Выполнила: студентка 6 курса
группы 22609 А. М. Луковникова

подпись

Руководитель курса:
к.ф-м.н., доцент К. А. Кулаков
Итоговая оценка:

подпись

Петрозаводск

2015

Содержание

1	Выбор и согласование объекта тестирования.	3
2	Архитектура системы.	4
3	План тестирования.	6
4	Тесты.	8
4.1	Блочные тесты.	8
4.2	Интеграционные тесты.	15
4.3	Нагрузочные тесты.	18
4.4	Аттестационные тесты.	19
5	Пример реализации тестов.	20
6	Результаты.	22
6.1	Результаты блочного тестирования.	22
6.2	Результаты интеграционного тестирования.	23
6.3	Результаты нагрузочного тестирования.	23
6.4	Результаты аттестационного тестирования.	24
6.5	Покрытие кода.	24
7	Протокол тестирования	25
8	Приложение 1.	27
	Библиографический список использованной литературы.	29

1 Выбор и согласование объекта тестирования.

Объектом тестирования была выбрана программа расчета железобетонных конструкций стен методом конечных элементов, реализуемая в рамках магистерской диссертации.

В настоящее время в строительстве для выполнения статических расчетов конструктивных элементов (определения усилий, напряжений и деформаций в элементах конструктивной схемы), как правило, используются специальные программные комплексы, в основе которых лежит метод конечных элементов (МКЭ).

WallCalculation – программа расчета железобетонных балок-стенок с учетом образования и развития трещин методом конечных элементов.

Данная программа позволяет пользователю вычислить характеристики, которые могут появиться при эксплуатации здания:

- напряжения в бетоне;
- напряжения в арматуре;
- напряжения в железобетоне;
- перемещения;
- трещины (угол наклона, если трещина появилась).

Помимо таблиц со значениями характеристик, описанных выше, пользователю предоставляется информация о том, как будет протекать процесс разрушения стены при увеличении эксплуатационных нагрузок.

2 Архитектура системы.

На Рис.1. представлена архитектура тестируемой системы.

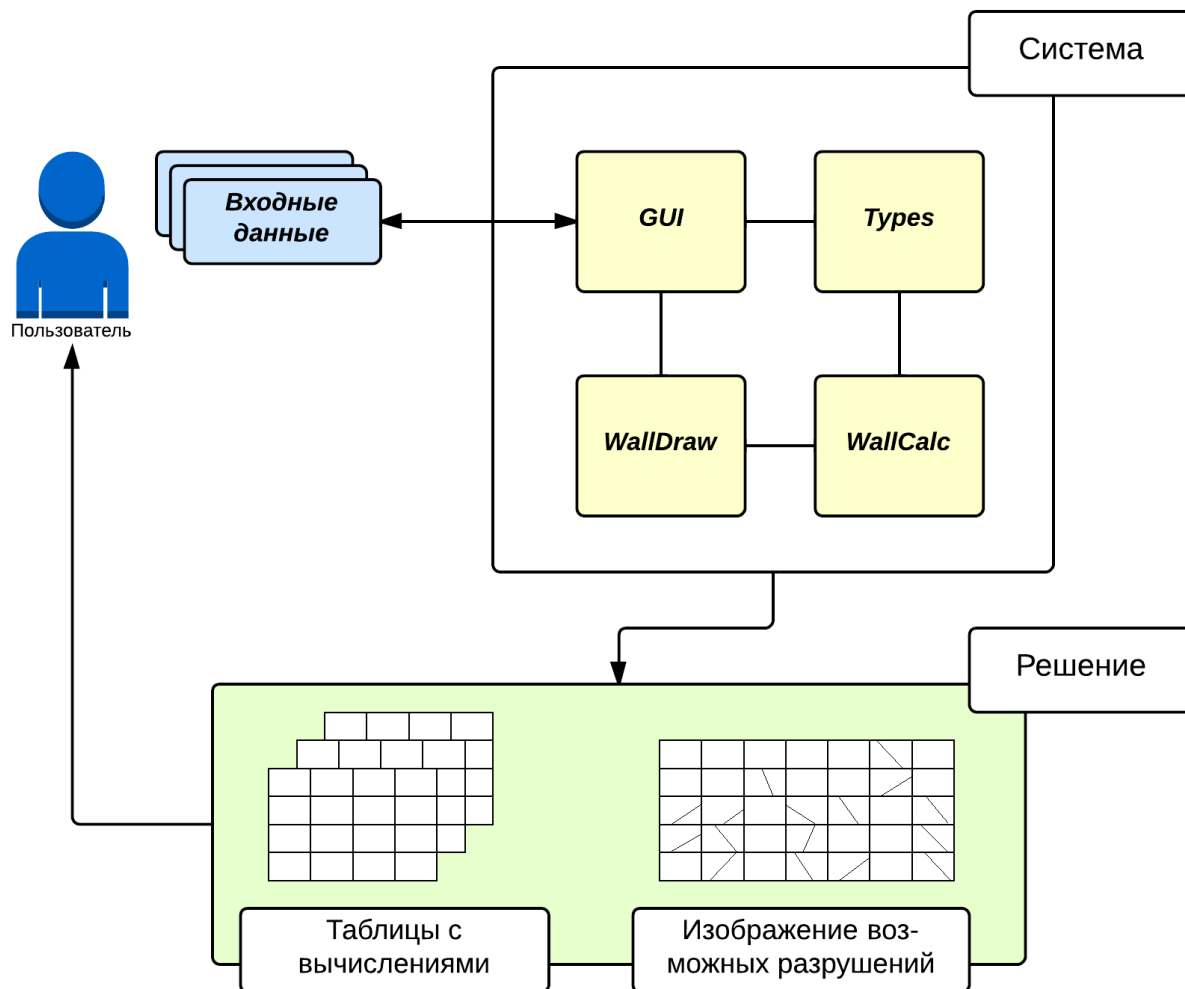


Рис. 1: Архитектура системы.

- Модуль GUI – графический интерфейс пользователя;
Предоставляет пользователю визуализированную информацию, описанную в Прил. 1.
- Types – модуль внутреннего представления данных;
Задача расчета железобетонной конструкции стены имеет множество входных параметров. Например, для стены разбитой на 120 КЭ файл с входными данными состоит из 442 строк. Поэтому было принято решение о выделении отдельного модуля, для реализации представления физических свойств КЭ и нагрузок в виде классов с соответствующими полями и методами. Модуль

Types так же отвечает за считывание входной информации (из входных файлов формата *.txt или графического интерфейса пользователя), а так же за предоставление этой информации другим модулям.

- WallDraw – модуль отрисовки данных;

Данный модуль выполняет отрисовку входных данных и данных, полученных после расчета, т.е. после работы модуля WallCalc. Модуль находится на стадии разработки.

- WallCalc – модуль поиска решения.

В данном модуле происходит решение задачи расчета железобетонных конструкций стен. Решение производится методом конечных элементов.

Решение, в виде таблиц с вычислениями (напряжения в бетоне, напряжения в арматуре, напряжения в железобетоне, трещины, перемещения) и изображений возможных разрушений, отображается в графическом интерфейсе пользователя, а так же сохраняется на компьютер.

3 План тестирования.

На Рис.2. показана диаграмма взаимодействия модулей системы.

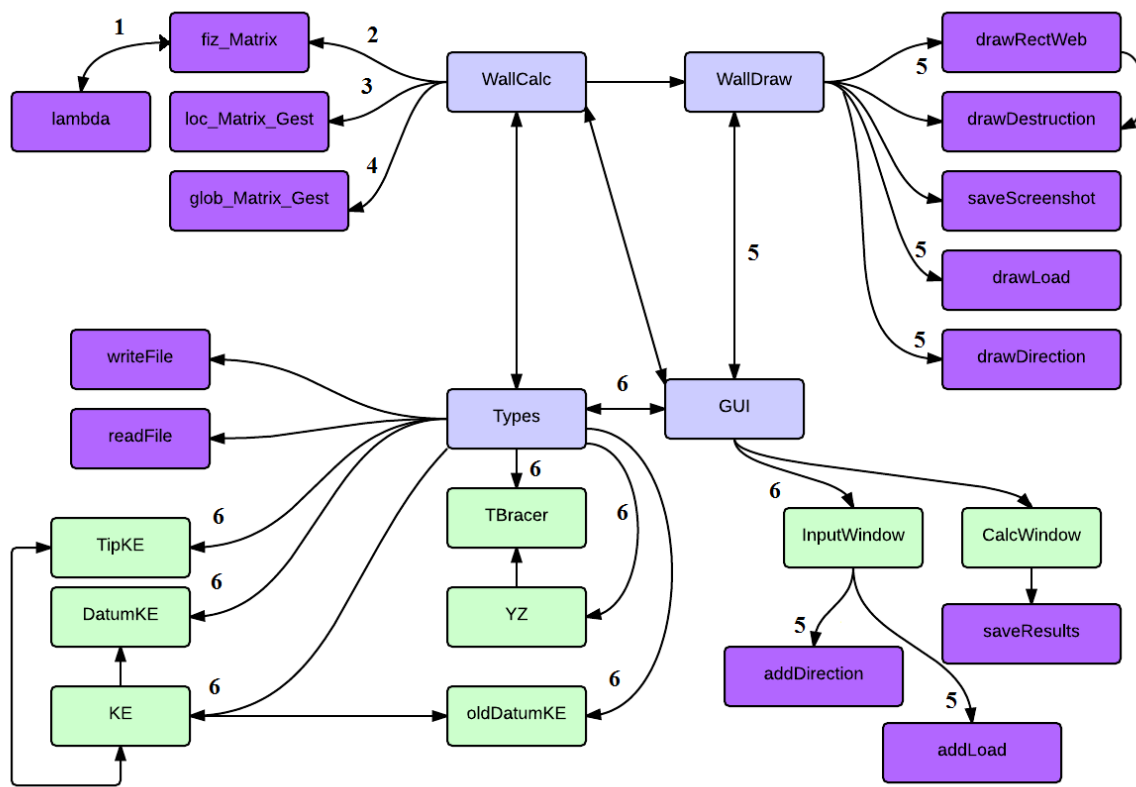


Рис. 2: Диаграмма взаимодействия модулей системы. Сиреневым цветом изображены модули системы, фиолетовым – функции, а зеленым структуры и классы.

Тестирование будут проходить следующие компоненты:

- WallCalc (модуль поиска решения).
 1. lambda – функция, которая вычисляет коэффициенты, учитывающие работу арматуры в трещинах на сдвиг и среднее расстояние между трещинами;
 2. fiz_Matrix – функция, которая формирует матрицы физических характеристик бетона и арматуры; Использует результаты работы функции lambda().
 3. loc_Matrix_Gest – функция формирования локальных матриц жесткости;
 4. glob_Matrix_Gest – функция формирования глобальных матриц жесткости.

- WallDraw (модуль отрисовки данных).
 1. drawRectWeb – функция отрисовки балки-стенки, разбитой на КЭ.
 2. drawDestruction – функция отрисовки балки-стенки с возможными разрушениями (после работы модуля WallCalc). Эта функция не будет проходить тестирование, т.к. не реализована до конца.
 3. saveScreenshot – функция сохранения изображения балки-стенки на текущем этапе работы программы. Эта функция не будет проходить тестирование, т.к. выполнена стандартными средствами Visual Studio.
 4. drawLoad – функция отрисовки приращения нагрузки по осям X,Y.
 5. drawDirection – функция отрисовки направления связей по осям X,Y.
- Types (модуль внутреннего представления данных).
 1. writeFile – запись в файл результатов работы программы. Эта функция не будет проходить тестирование, т.к. выполнена стандартными средствами при помощи сериализации.
 2. readFile – чтение из входного файла. Эта функция не будет проходить тестирование, т.к. выполнена стандартными средствами при помощи сериализации.
 3. TipKE – тип КЭ (бетон, железобетон, пустой, опорный элемент).
 4. DatumKE – напряжения, деформации КЭ и перемещения в узлах КЭ.
 5. KE – свойства КЭ. Основная структура программы.
 6. TBracer – наличие связи в узле.
 7. YZ – список информации об узлах КЭ.
 8. oldDatumKE – напряжения и деформации на предыдущей итерации.
- GUI (графический интерфейс пользователя).
 1. InputWindow – окно отрисовки входных данных.
 - addDirection – добавление направления связей по осям X,Y.
 - addLoad – добавление приращения нагрузки по осям X,Y.
 2. CalcWindow – окно отрисовки результатов расчетов.
 - saveResults – функция сохранения результатов вычислений. Эта функция не будет проходить тестирование, т.к. не реализована до конца.

4 Тесты.

Блочное тестирование будет применено ко всем функциям модуля WallCalc (см. 2, 3, 4 на Рис.2.), описанным в параграфе 3 данного отчета, кроме функции `fiz_Matrix`. Требования для проведения блочного тестирования: ОС Windows, Visual Studio 2010.

Интеграционное тестирование (нисходящее) будет проходить в следующем порядке:

1. функции `int fiz_Matrix()` , `int lambda(KE Element, double alpha)` Тестируется взаимодействие двух функций модуля WallCalc. Функция `fiz_Matrix()` использует результат работы функции `lambda()` (см. 1 на Рис.2.).
2. `InputWindow -> TipKE, DatumKE, KE, TBracer, YZ, oldDatumKE`. Тестируется взаимодействие модулей GUI и Types. Корректность заполнения структур программы через графический интерфейс пользователя (см. 6 на Рис.2.).
3. `InputWindow -> drawRectWeb, drawLoad, drawDirection` Тестируется взаимодействие модулей GUI и WallDraw. Корректность отрисовки входных данных (см. 5 на Рис.2.).

Нагрузочное тестирование будет проходить модуль WallDraw (`drawRectWeb`). Так как решение задачи железобетона зависит от количества КЭ, на которые разбита балка- стенка, то важной проблемой реализации расчета железобетонных конструкций стен является визуализация стены, для которой выполняется расчет. Чем больше КЭ, на которые разбита стена, тем ближе результаты работы программы к теоретическим, однако это увеличивает нагрузку на систему. Поэтому принято решение протестировать время отклика системы. Для отрисовки стены 100x100 система не должна тратить более 60 секунд.

Аттестационное тестирование будет проходить система в целом, для того, чтобы проверить взаимодействие всех компонент.

4.1 Блочные тесты.

Функции `lambda`, `fiz_Matrix`, `loc_Matrix_Gest`, `glob_Matrix_Gest` работают со структурой KE. Данная структура содержит следующие поля:

1. `int L` – длина КЭ. Принимает значения $L > 0$.
2. `int H` – высота КЭ. Принимает значения $H > 0$.
3. `int Number` – номер КЭ. Принимает значения $1 \leq Number \leq L * H$.
4. `TipKE Tip` – тип КЭ. Принимает значения "Бетон" "Железобетон" "Опорный элемент" "Пустой". В зависимости от этого значения элементу присваиваются различные свойства, описанные в параграфе 2. Эти свойства хранятся в классе `TipKE`.
5. `Matrix Db` – матрица физ. характеристик бетона. Принимает значения в диапазоне $\pm 5.0 \times 10^{-324}$ до $\pm 1.7 \times 10308$. Размерность матрицы 3x3.
6. `Matrix Ds` – матрица физ. характеристик арматуры. Принимает значения в диапазоне $\pm 5.0 \times 10^{-324}$ до $\pm 1.7 \times 10308$. Размерность матрицы 3x3.
7. `Matrix D` – матрица физ. характеристик железобетона. Принимает значения в диапазоне $\pm 5.0 \times 10^{-324}$ до $\pm 1.7 \times 10308$. Размерность матрицы 3x3.
8. `Matrix oldD` – матрица физ. характеристик ж/бетона на предыдущей итерации. Принимает значения в диапазоне $\pm 5.0 \times 10^{-324}$ до $\pm 1.7 \times 10308$. Размерность матрицы 3x3.
9. `Matrix K` – матрица жесткости КЭ. Принимает значения в диапазоне $\pm 5.0 \times 10^{-324}$ до $\pm 1.7 \times 10308$. Размерность матрицы 8x8.
10. `List<List<int>> Yzl` – индексы узлов, которые примыкают к КЭ. Принимают значения $[M*Number, i*M*Number+1, M*Number+N+1, M*Number+N+2]$.
 $M, N \geq 0$.
11. `DatumKE Data` – напряжения и деформации КЭ и перемещения в узлах КЭ. Принимает значения 'U', 'V', 'X', 'Y'. В зависимости от этого значения элементу присваиваются различные свойства.
12. `oldDatumKE oldData` – напряжения и деформации на предыдущей итерации. Принимает значения 'U', 'V', 'X', 'Y'. В зависимости от этого значения элементу присваиваются различные свойства.

int lambda(KE Element, double alpha)

Тест: 1

Тип теста: общий

Описание: тест проверяет правильность работы функции на корректных значениях KE Element (описание корректных значений приведено выше) и alpha.

Входные данные: KE Element заполнен верно, $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка.

Ожидаемый результат: return 0.

Тест: 2

Тип теста: негативный

Описание: тест проверяет правильность работы функции на корректных значениях KE Element и некорректных значениях alpha.

Входные данные: KE Element заполнен верно, $alpha = 0$ или делится на $\frac{\pi}{2}$ без остатка.

Ожидаемый результат: return -1.

Тест: 3

Тип теста: краевой

Описание: тест проверяет правильность работы функции назначениях alpha близких к 0 или к $\frac{\pi}{2}$.

Входные данные: KE Element заполнен верно, $alpha = \frac{1}{10^{16}}$.

Ожидаемый результат: return 0.

Тест: 4

Тип теста: краевой Описание: тест проверяет правильность работы функции на значениях L и H близких к 0.

Входные данные: KE Element заполнен верно, $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка, $L, H \rightarrow 10^{-16}$.

Ожидаемый результат: return 0.

Тест: 5

Тип теста: простой

Описание: тест проверяет правильность работы функции если стена разбита на 1

КЭ.

Входные данные: KE Element заполнен верно, $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка, $Yzl = [1,2,3,4]$.

Ожидаемый результат: return 0.

Тест: 6

Тип теста: негативный

Описание: тест проверяет работу функции если L или H = 0.

Входные данные: $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка, H = 0.

Ожидаемый результат: return -2.

Тест: 7

Тип теста: негативный

Описание: тест проверяет работу функции если L или H = 0.

Входные данные: $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка, L = 0.

Ожидаемый результат: return -2.

Тест: 8

Тип теста: негативный

Описание: тест проверяет работу функции если Tip имеет некорректное значение.

Входные данные: $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка, Tip = "бетонпустой".

Ожидаемый результат: return -3.

Тест: 9

Тип теста: негативный

Описание: тест проверяет работу функции если H || L < 0.

Входные данные: $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка, H || L < 0.

Ожидаемый результат: return -4.

Тест: 10

Тип теста: негативный

Описание: тест проверяет работу функции если у разных КЭ одинаковые индексы.

Входные данные: $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка, $Yzl_1 = [1,2,3,4]$ и $Yzl_2 = [1,2,3,4]$.

Ожидаемый результат: return -5.

Тест: 11

Тип теста: негативный

Описание: тест проверяет работу функции если у одного КЭ есть отрицательный индекс.

Входные данные: $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка, $Yz1_1 = [1,2,-28,4]$.

Ожидаемый результат: return -6.

Тест: 12

Тип теста: краевой

Описание: тест проверяет работу функции если H и $L \rightarrow 2147483647$.

Входные данные: $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка, H и $L = 2\ 147\ 483\ 641$.

Ожидаемый результат: return 0.

Тест: 13

Тип теста: негативный

Описание: тест проверяет работу функции если КЕ пуст.

Входные данные: $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка, $KE=Null$.

Ожидаемый результат: return -8.

Тест: 14

Тип теста: негативный

Описание: тест проверяет работу функции если Db , $Ds,oldD$ или D имеют размерность не 3×3 .

Входные данные: $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка, Db , $Ds,oldD$ или D имеют размерность 3×4 .

Ожидаемый результат: return -9.

Тест: 15

Тип теста: негативный

Описание: тест проверяет работу функции если K имеет размерность не 8×8 .

Входные данные: $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка, K имеет размерность 8×7 .

Ожидаемый результат: return -10.

Тест: 16

Тип теста: негативный

Описание: тест проверяет работу функции если Data или oldData имеют некорректные значения.

Входные данные: $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка, Data или oldData имеет значение = "UX".

Ожидаемый результат: return -11.

`int loc_Matrix_Gest()`

Тест: 17

Тип теста: общий

Описание: тест проверяет правильность работы функции на корректных значениях KE Element (описание корректных значений приведено выше).

Входные данные: KE Element заполнен верно.

Ожидаемый результат: return 0.

Тест: 18

Тип теста: негативный

Описание: тест проверяет работу функции если $H \parallel L < 0$.

Входные данные: $H \parallel L < 0$.

Ожидаемый результат: return -4.

Тест: 19

Тип теста: негативный

Описание: тест проверяет работу функции если H и $L \rightarrow 2147483647$.

Входные данные: H и $L = 2\ 147\ 483\ 641$.

Ожидаемый результат: return 0.

Тест: 20

Тип теста: негативный

Описание: тест проверяет работу функции если K имеет размерность не 8×8 .

Входные данные: K имеет размерность 8×7 .

Ожидаемый результат: return -10.

Тест: 21

Тип теста: негативный

Описание: тест проверяет работу функции если К имеет значения не входящие в диапазон значений.

Входные данные: К имеет элемент 10000000000000000.

Ожидаемый результат: return -12.

int glob_Matrix_Gest()

Тест: 22

Тип теста: общий

Описание: тест проверяет правильность работы функции на корректных значениях KE Element (описание корректных значений приведено выше).

Входные данные: KE Element заполнен верно.

Ожидаемый результат: return 0.

Тест: 23

Тип теста: негативный

Описание: тест проверяет работу функции если у разных КЭ одинаковые индексы.

Входные данные: Yzl_1 = [1,2,3,4] и Yzl_2 = [1,2,3,4].

Ожидаемый результат: return -5.

Тест: 24

Тип теста: негативный

Описание: тест проверяет работу функции если у одного КЭ есть отрицательный индекс.

Входные данные: Yzl_1 = [1,2,-28,4].

Ожидаемый результат: return -6.

Тест: 25

Тип теста: негативный

Описание: тест проверяет работу функции если у одного КЭ неправильно посчита-

ны индексы узлов.

Входные данные: $Yzl_1 = [1,3,4,9]$.

Ожидаемый результат: return -7.

Тест: 26

Тип теста: негативный

Описание: тест проверяет работу функции если Data или oldData имеют некорректные значения.

Входные данные: Data или oldData имеет значение = "UX".

Ожидаемый результат: return -11.

Тест: 27

Тип теста: простой

Описание: тест проверяет работу функции если Data или oldData имеют корректные значения.

Входные данные: Data или oldData имеет значение = "X".

Ожидаемый результат: return 0.

4.2 Интеграционные тесты.

`int fiz_Matrix() -> int lambda(KE Element, double alpha)`

Тест: 28

Тип теста: простой

Описание: тест проверяет правильность работы функций если стена разбита на 1 КЭ.

Входные данные: KE Element заполнен верно, $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка, $Yzl = [1,2,3,4]$.

Ожидаемый результат: return 0.

Тест: 29

Тип теста: общий

Описание: тест проверяет правильность работы функций если стена разбита на 10

КЭ.

Входные данные: KE Element заполнен верно, $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка.

Ожидаемый результат: return 0.

Тест: 30

Тип теста: негативный

Описание: тест проверяет работу функций если KE пуст.

Входные данные: $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка, KE=Null.

Ожидаемый результат: return -8.

Тест: 31

Тип теста: негативный

Описание: тест проверяет работу функций если имеется 1 KE, который заполнен неверно.

Входные данные: $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка, у одного KE

Tip="железобетонжелезобетон".

Ожидаемый результат: return -17.

Тест: 32

Тип теста: негативный

Описание: тест проверяет работу функций если имеется 2 KE с одинаковыми индексами.

Входные данные: $alpha \neq 0$ и не делится на $\frac{\pi}{2}$ без остатка, Yzl_1 = [1,2,3,4] и Yzl_2 = [1,2,3,4].

Ожидаемый результат: return -5.

Тест: 33

Тип теста: негативный

Описание: тест проверяет правильность работы функций на корректных значениях KE Element и некорректных значениях alpha.

Входные данные: KE Element заполнен верно, $alpha = 0$ или делится на $\frac{\pi}{2}$ без остатка.

Ожидаемый результат: return -1.

InputWindow -> TipKE, DatumKE, KE

Тест: 34

Тип теста: общий

Описание: тест проверяет правильность работы программы на корректных значениях.

Входные данные: KE Element заполнен верно

Ожидаемый результат: структуры заполнены.

Тест: 35

Тип теста: негативный

Описание: тест проверяет работу программы, если введено очень большое значение приращения нагрузки.

Входные данные: KE Element заполнен верно, Iteraz (приращение нагрузки)=10000000000.

Ожидаемый результат: вывод предупреждения, продолжение работы программы.

Тест: 36

Тип теста: негативный

Описание: тест проверяет работу программы, если произойдет изменение свойств какого-либо КЭ.

Входные данные: KE Element заполнен верно, коэффициент Пуассона=2 (должен быть 0.2).

Ожидаемый результат: вывод предупреждения, продолжение работы программы.

InputWindow -> drawRectWeb, drawLoad, drawDirection

Тест: 37

Тип теста: общий

Описание: тест проверяет правильность работы функции на корректных значениях KE Element (описание корректных значений приведено выше).

Входные данные: KE Element заполнен верно

Ожидаемый результат: выполняется отрисовка стенки, разбитой на КЭ с элементами нагрузки и направлений.

Тест: 38

Тип теста: простой

Описание: тест проверяет правильность работы функции на 1 КЭ

Входные данные: КЕ Element заполнен верно

Ожидаемый результат: выполняется отрисовка 1 КЭ с элементами нагрузки и направлений..

Тест: 39

Тип теста: специальный

Описание: тест проверяет работу функций, если пользователь пытается сначала задать свойства, а потом нагрузку

Входные данные:

Ожидаемый результат: вывод предупреждения, продолжение работы программы.

Тест: 40

Тип теста: негативный

Описание: тест проверяет работу функций, если H или $L = 0$.

Входные данные: H или $L = 0$.

Ожидаемый результат: вывод предупреждения, продолжение работы программы.

4.3 Нагрузочные тесты.

Так как точность решения задачи железобетона зависит от числа КЭ, на которые разбита балка-стенка, было принято решение провести нагрузочное тестирование модуля WallDraw. Главный параметр для оценки работы модуля – это время, которое требуется модулю для отрисовки стены, разбитой на КЭ.

В процессе тестирования выполнялась следующая последовательность действий:

1. запуск программы;
2. задание критических значений (см. Прим.1);

3. оценка времени работы модуля WallDraw;
4. завершение программы.

Примечание №1: под критическими понимаются большие значения начальных данных (т.е. большое число КЭ), которые могут привести к некорректной работе программы (например, зависание системы).

4.4 Аттестационные тесты.

- Аттестационное тестирование программы на произвольных данных.

Действия:

- Запуск программы.
- Задание начальных условий.
- Задание граничных условий.
- Задание свойств КЭ.
- Задание нагрузки.
- Расчет.

Ожидаемый результат:

- Запуск должен осуществиться.
- Задание начальных и граничных условий происходит корректно.
- Задание свойств КЭ и нагрузки происходит корректно.
- Расчет осуществляется корректно.
- При обнаружении ошибок в начальных данных выводится сообщение об ошибке.

- Аттестационное тестирование на теоретических данных.

- Сценарий аналогичный, однако ожидается отсутствие ошибок в начальных данных, а так же соответствие результатов расчета программы теоретическим результатам.

5 Пример реализации тестов.

Для блочного тестирования было выбрано средство Unit Test VS2010. Функция `loc_Matrix_Gest()` работает со статической локальной матрицей КЭ. В процессе работы она заполняет локальные матрицы жесткости для каждого элемента. В результате функция возвращает:

- `true`, если матрицы заполнены;
- `false`, если матрицы не заполнены, т.е. произошла какая-то ошибка.

Тест, представленный на Listing 1, принимает на вход значения:

- `N`, `M` – размер матрицы КЭ;
- `Result` – ожидаемый результат работы функции (`true`, `false`).

После считывания входных данных инициализируется матрица КЭ случайными значениями, используя библиотеку `AutoFixture`. Затем вызывается функция `Assert.AreEqual(result, CalcModule_Accessor.loc_Matrix_Gest())`, которая сравнивает ожидаемый результат из Listing 2 с полученным. Так же корректность работы функции `loc_Matrix_Gest()` проверяется с помощью `Assert.IsNotNull(Element.K[k, l])` были ли присвоены значения локальной матрице жесткости для каждого элемента.

```
[DataSource("Microsoft.VisualStudio.TestTools.DataSource.XML",
    "|DataDirectory|\\InputTestData.xml", "Test", DataAccessMethod.Sequential),
    DeploymentItem("TestProject1\\InputTestData.xml"), TestMethod()]
public void loc_Matrix_GestTest()
{
    int N = int.Parse(TestContext.DataRow["N"].ToString());
    int M = int.Parse(TestContext.DataRow["M"].ToString());
    bool result = bool.Parse(TestContext.DataRow["Result"].ToString());

    var fixture = new Fixture();
    CalcModule_Accessor.KEList = new ObservableCollection<KE>(fixture.CreateMany<KE>(N *
        M));

    Assert.AreEqual(result, CalcModule_Accessor.loc_Matrix_Gest());

    for(int i=0; i<N; i++)
        for (int j = 0; j < M; j++)
            {
```

```
    KE Element = CalcModule_Accessor.KEList[i * j];

    for (int k = 1; k <= 8; k++)
        for (int l = 1; l <= 8; l++)
        {
            Assert.IsNotNull(Element.K[k, l]);
        }
    }
}
```

Listing 1: Пример теста для функции loc_Matrix_Gest()

6 Результаты.

6.1 Результаты блочного тестирования.

Таблица 1: Результаты блочного тестирования.

Функция	Количество тестов	Количество ошибок
lambda()	16	6 (тесты: 3, 6, 7, 10, 11, 12)
loc_Matrix_Gest()	5	1 (тесты: 19)
int glob_Matrix_Gest	6	2 (тесты: 24, 25)

В Таблице 1 приведены результаты блочного тестирования.

Описание ошибок:

1. Тест: 3. return -31. Деление на 0.
2. Тест: 6. return 0. Нет проверки на L или $H \neq 0$
3. Тест: 7. return 0. Нет проверки на L или $H \neq 0$
4. Тест: 10. return 0. Нет проверки на корректность значений индексов соседних узлов.
5. Тест: 11. return 0. Нет проверки на корректность значений индексов соседних узлов.
6. Тест: 12. return -15. Не работает на очень больших значениях.
7. Тест: 19. return -15. Не работает на очень больших значениях.
8. Тест: 24. return 0. Нет проверки на корректность значений индексов соседних узлов.
9. Тест: 25. return 0. Нет проверки на корректность значений индексов соседних узлов.

Таблица 2: Результаты интеграционного тестирования.

Функция	Кол-во тестов	Кол-во ошибок
int fiz_Matrix() -> int lambda()	6	1 (тесты: 32)
InputWindow -> TipKE, DatumKE, KE	3	2 (тесты: 35, 36)
InputWindow -> drawRectWeb, ...	4	2 (тесты: 39, 40)

6.2 Результаты интеграционного тестирования.

В Таблице 2 приведены результаты интеграционного тестирования.

Описание ошибок:

1. Тест: 32. return 0. Нет проверки на корректность значений индексов соседних узлов.
2. Тест: 35. Не выводит предупреждение, продолжает работу.
3. Тест: 36. Не выводит предупреждение, продолжает работу.
4. Тест: 39. Не выводит предупреждение, продолжает работу.
5. Тест: 40. Не выводит предупреждение, продолжает работу.

6.3 Результаты нагрузочного тестирования.

Отчет о ходе выполнения нагрузочного тестирования приведен в Таблице 3.

Таблица 3: Результаты нагрузочного тестирования.

Количество КЭ	Время	Результат
100*10	< 3 сек.	Успех.
20*100	< 10 сек.	Успех.
50*50	< 19 сек.	Успех.
50*40	< 10 сек.	Успех.

6.4 Результаты аттестационного тестирования.

Отчет о ходе выполнения аттестационного тестирования будет приведен в протоколе тестирования.

6.5 Покрытие кода.

Расчет тестового покрытия относительно исполняемого кода программного обеспечения проводится по формуле:

$$T_{cov} = \frac{L_{tc}}{L_{code}} * 100\%$$

где:

T_{cov} – тестовое покрытие;

L_{tc} – кол-ва строк кода, покрытых тестами;

L_{code} – общее количество строк кода.

В данном случае:

$$T_{cov} = \frac{L_{tc}}{L_{code}} * 100\% = \frac{1937}{2389} * 100\% = 81\%$$

7 Протокол тестирования

№	Тест (id или описание)	Результат
1	Аттест., на теорет. данных	Успех
2	Аттест., на теорет. данных (2)	Успех
3	Аттест., на произв. данных	Успех
4	Аттест., на произв. данных (2)	Успех
5	Нагрузочный, 100*10 КЭ	Успех
6	Нагрузочный, 20*100 КЭ	Успех
7	Нагрузочный, 50*50 КЭ	Успех
8	Нагрузочный, 50*40 КЭ	Успех
9	1	Успех
10	2	Успех
11	3	Ош. Деление на 0
12	4	Успех
13	5	Успех
14	6	Ош. Нет проверки $L H \neq 0$
15	7	Ош. Нет проверки $L H \neq 0$
14	6	Успех
15	7	Успех
16	8	Успех
17	9	Успех
18	10	Ош. Неправильно посчитаны узлы
19	11	Ош. Неправильно посчитаны узлы
20	12	Ош. Не работает с большими значениями
21	13	Успех
22	14	Успех
23	15	Успех
24	16	Успех
25	17	Успех
26	18	Успех
27	19	Ош. Не работает с большими значениями

№	Тест (id или описание)	Результат
28	20	Успех
29	21	Успех
30	22	Успех
31	23	Успех
32	24	Ош. Неправильно посчитаны узлы
33	25	Ош. Неправильно посчитаны узлы
34	26	Успех
35	27	Успех
36	28	Успех
37	29	Успех
38	30	Успех
39	31	Успех
40	32	Ош. Неправильно посчитаны узлы
41	33	Успех
42	34	Успех
43	35	Не выводит предупреждение.
44	36	Не выводит предупреждение.
45	37	Успех
46	38	Успех
47	39	Не выводит предупреждение.
48	40	Не выводит предупреждение.
49	35	Успех
50	36	Успех
51	39	Успех
52	40	Успех

8 Приложение 1.

- Задание начальных данных:

1. количество столбцов КЭ;
2. количество строк КЭ;
3. размеры КЭ (высота, длина).

- Изображение балки-стенки, разбитой на КЭ.

- Задание граничных условий (направление связей по осям X, Y).

- Задание свойств КЭ;

В зависимости от типа КЭ (бетон, железобетон, опорный элемент или пустой) происходит присвоение различных значений следующих свойств:

1. начальный модуль упругости бетона;
2. коэффициент Пуассона;
3. расчетное сопротивление сжатию бетона;
4. расчетное сопротивление растяжению бетона;
5. нормативное сопротивление растяжению бетона;
6. модуль упругости арматуры;
7. предел упругой работы арматуры;
8. расчетное сопротивление сжатию арматуры;
9. сопротивление разрыва арматуры;
10. коэффициенты армирования по осям X, Y;
11. диаметры армирования по осям X, Y;
12. цвет КЭ.

- Толщина балки-стенки.

- Приращение нагрузки по осям X, Y.

Данная информация считывается из входных файлов, либо вводится пользователем самостоятельно.

- Изображение деформированной балки-стенки.

- Отображение таблиц с расчетами:

1. напряжения в бетоне по осям X,Y;
2. напряжения в арматуре по осям X,Y;
3. напряжения в железобетоне по осям X,Y;
4. перемещения по осям X,Y;
5. трещины (угол наклона, если трещина появилась).

Эта информация отображается после работы модулей WallDraw и WallCalc.

- Отчеты об ошибках во входных данных.

Список литературы

1. Петров А. Н., Воронин З. А. *Расчет железобетонных конструкций стен методом конечных элементов. Часть I. Линейно-упругий расчет* Петрозаводск: Издательство ПетрГУ. 2010. — 64 с.
2. Воронин З. А. *Деформационная модель и методика расчета железобетонных балок-стенок с учетом образования и развития трещин* Петрозаводск: Издательство ПетрГУ. 2009. — 31 с.
3. *Пошаговое руководство. Создание и запуск модульных тестов для управляемого кода* [Электронный ресурс]/электрон.ст., 2014.- Режим доступа к ст.:<http://msdn.microsoft.com/ru-ru/library/ms182532.aspx>, свободный.- Загл. с экрана.- Яз.рус.- (Дата обращения: 15.11.2014)