



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
*«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)*

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ,

информационные технологии»

ДОМАШНЯЯ РАБОТА

«МОДУЛЬНОЕ ТЕСТИРОВАНИЕ»

ДИСЦИПЛИНА: «Тестирование и отладка программного обеспечения»

Выполнил: студент гр. ИУК4-82Б _____ (_____ Панина В.С. _____)
(подпись) (Ф.И.О.)

Проверил: _____ (_____ Красавин Е.В. _____)
(подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2021

Цель: формирование у студентов практических навыков составления набора тестовых данных для модульного тестирования.

Задачи:

1. Провести обзор разработанного программного кода.
2. Для каждого модуля построить графы и вычислить цикломатические числа. Разработать тестовые случаи для каждого графа
3. Провести модульное тестирование согласно составленным тестовым случаям

Задание:

1. Провести обзор разработанного программного кода. В случае обнаружения ошибок составить отчет следующего содержания (табл. 1).

Таблица 1

Ошибки обзора Номер ошибки	Название модуля/ функции	Описание ошибки	Важность ошибки (высокая, средняя, низкая)	Ошибка исправлена Да/Нет
1	validateForm()	Неправильное присвоение статуса валидности	Средняя	Да
2	Проверка валидности логина	Неправильное составление регулярного выражения	Средняя	Да
3	Проверка существования пользователя с таким логином	Неправильно составление SQL запроса	Высокая	Да
4	Добавление нового пользователя в БД	Неправильно составление SQL запроса	Высокая	Да
5	Авторизация пользователя	Неправильное составление условий проверки	Высокая	Да
6	Вывод каталога товаров	Некорректное указание индексов выводимых полей	Высокая	Да

7	add_to_basket()	Неправильно составление SQL запроса	Высокая	Да
8	Вывод корзины	Неправильно составление SQL запроса	Средняя	Да
9	Выход из учетной записи	Не происходит выход из учетной записи	Низкая	Нет

2. Для каждого модуля построить графы и вычислить цикломатические числа. Разработать тестовые случаи для каждого графа и представить их в виде табл. 2.

Метрический показатель сложности или цикломатическое число G потокового графа определяется по формуле:

$$G = R - V + 2,$$

где R – количество ребер графа;

V – количество вершин графа.

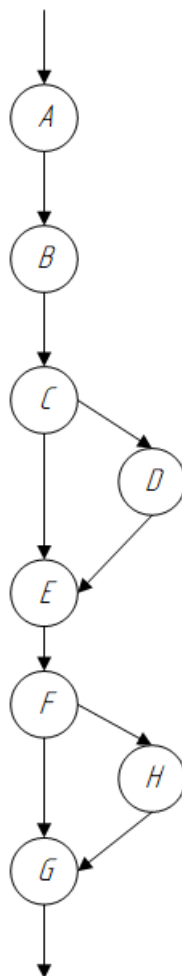


Рис. 1. Модуль ValidateForm()

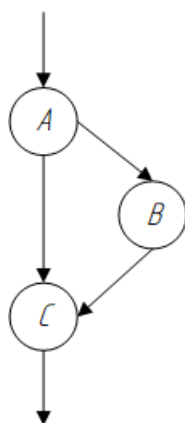


Рис. 2. Модуль Проверка валидности логина

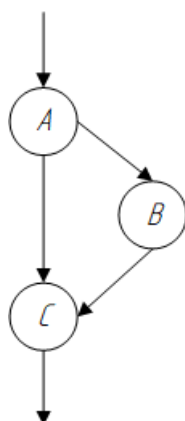


Рис. 3. Модуль проверки существования пользователя с таким логином

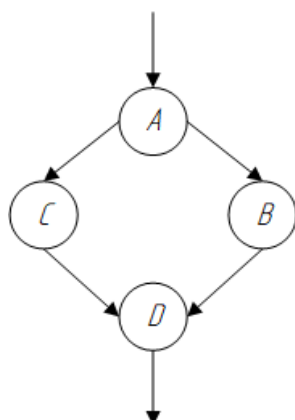


Рис. 4. Модуль добавления нового пользователя в БД

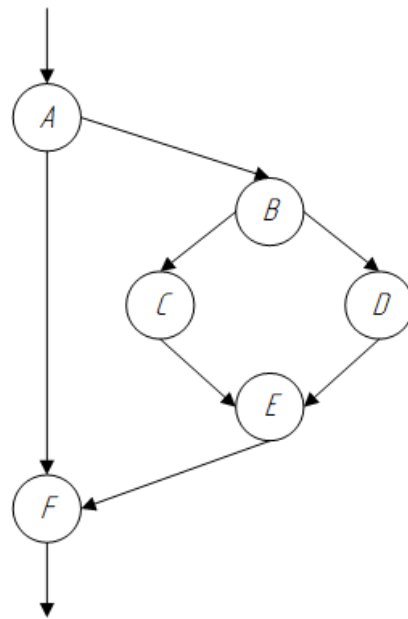


Рис. 5. Модуль авторизации пользователя

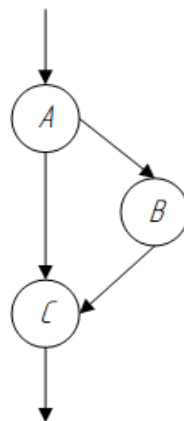


Рис. 6. Модуль вывода каталога товаров



Рис. 7. Модуль add_to_basket()

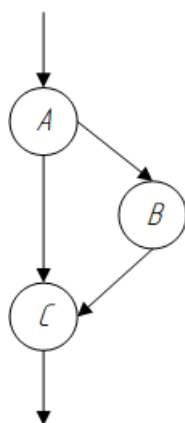


Рис. 8. Модуль вывода корзины

Таблица 2

Модуль validateForm () G	Номер сценария	Описание прохода	Контрольные примеры, позволяющие реализовать описанную ситуацию	Тест пройден Да/Нет
G = 4	1	a-b-c-d-e-f-h-g	y[i].value == "", valid == True	Да
	2	a-b-c-e-f-h-g	y[i].value == "a", valid == True	Да
	3	a-b-c-e-b-f-g	y[i].value == "", valid == False	Да
	4	a-b-c-e-f-g	y[i].value == "a", valid == False	Да
Модуль проверки валидности логина G	Номер сценария	Описание прохода	Контрольные примеры, позволяющие реализовать описанную ситуацию	Тест пройден Да/Нет
G = 2	1	a-b-c	\$_POST['login'] == «ФЫВВ»	Да
	2	a-c	\$_POST['login']= = «fsvr»	Да
Модуль проверки существован ия пользователя с таким логином G	Номер сценария	Описание прохода	Контрольные примеры, позволяющие реализовать описанную ситуацию	Тест пройден Да/Нет

G = 2	1	a-b-c	mysqli_num_rows(\$query) > 0	Да
	2	a-c	mysqli_num_rows(\$query) < 0	Да
Модуль добавления нового пользователя в БД G	Номер сценария	Описание прохода	Контрольные примеры, позволяющие реализовать описанную ситуацию	Тест пройден Да/Нет
G = 2	1	a-b-d	count(\$err) == 0	Да
	2	a-c-d	count(\$err) != 0	Да
Модуль авторизации пользователя G	Номер сценария	Описание прохода	Контрольные примеры, позволяющие реализовать описанную ситуацию	Тест пройден Да/Нет
G = 3	1	a-b-c-e-f	isset(\$_COOKIE['id']) and isset(\$_COOKIE['hash']) == True, (\$userdata['user_hash'] != \$_COOKIE['hash']) or (\$userdata['user_id'] != \$_COOKIE['id']) == True	Да
	2	a-b-d-e-f	isset(\$_COOKIE['id']) and isset(\$_COOKIE['hash']) == True, (\$userdata['user_hash'] != \$_COOKIE['hash']) or (\$userdata['user_id'] != \$_COOKIE['id']) == False	Да
	3	a-f	isset(\$_COOKIE['id']) and	Да

			isset(\$_COOKIE['hash']) == False	
Модуль вывода каталога товаров G	Номер сценария	Описание прохода	Контрольные примеры, позволяющие реализовать описанную ситуацию	Тест пройден Да/Нет
G = 2	1	a-b-c	\$result == True	Да
	2	a-c	\$result == False	Да
Модуль add_to_basket() G	Номер сценария	Описание прохода	Контрольные примеры, позволяющие реализовать описанную ситуацию	Тест пройден Да/Нет
G = 1	1	a-b-c	\$count=1; \$item_id = \$_POST['item_id'];	Да
Модуль вывода корзины G	Номер сценария	Описание прохода	Контрольные примеры, позволяющие реализовать описанную ситуацию	Тест пройден Да/Нет
G = 2	1	a-b-c	\$result == True	Да
	2	a-c	\$result == False	Да

3. Построить схему взаимодействия модулей.

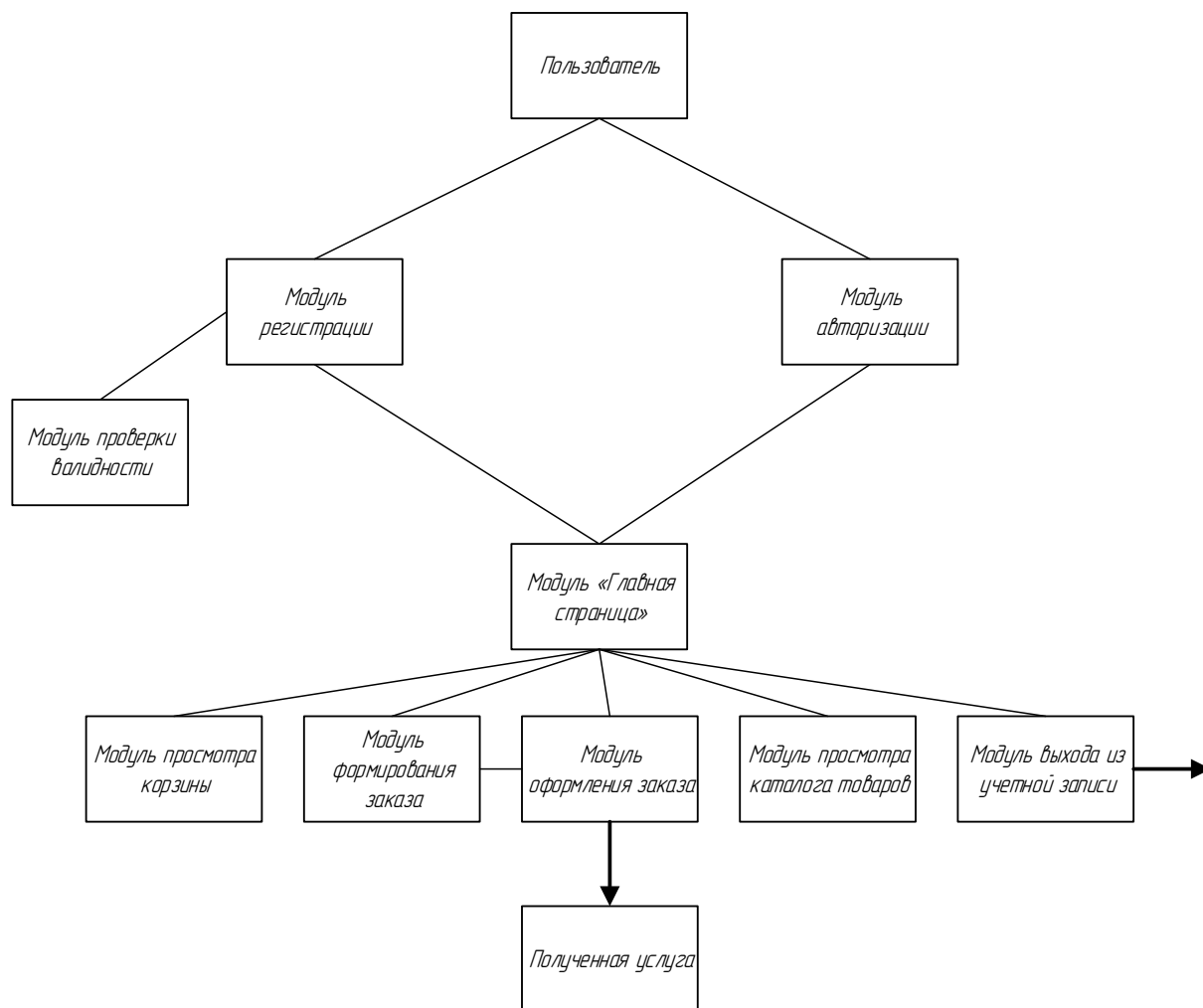


Рис. 9. Схема взаимодействия модулей

4. Разработать стратегию тестирования взаимодействия модулей и привести ее в виде табл. 3.

Таблица 3

Номер в последовательности	Описание последовательности	Контрольные примеры, позволяющие реализовать описанную ситуацию	Тест пройден Да/Нет
1	login_form() -> registration() -> check_validation()	register_button=True	Да
2	login_form() -> authorization()	authorization_button = True	Да
3	registration() -> main()	registration = True, check = True	Да
4	authorization() -> main()	authorization = True	Да

5	main() -> show_basket()	user=True	Да
6	main() -> order_formation()	user=True, basket != NULL	Да
7	main() -> order_formation() -> ordering()	user=True, basket != NULL, order_formation = true	Да
8	main() -> order_formation() -> ordering() -> received_order()	user=True, basket != NULL, order_formation = true, delivery_status = «delivered»	Да
9	main() -> show_items()	user=True	Да
10	main() -> exit()	user=True	Нет

Выводы: в результате выполнения домашней работы были сформированы практические навыки составления набора тестовых данных для модульного тестирования.

ПРИЛОЖЕНИЕ

Модуль validateForm():

```
function validateForm() {
    var x, y, i, valid = true;
    x = document.getElementsByClassName("tab");
    y = x[currentTab].getElementsByTagName("input");
    for (i = 0; i < y.length; i++) {
        if (y[i].value == "") {
            y[i].className += " invalid";
            valid = false;
        }
    }
    if (valid) {
        document.getElementsByClassName("step")[currentTab].className += "
finish";
    }
    return valid; }
```

Модуль проверки логина, существования пользователя, добавления нового пользователя в БД:

```
if(isset($_POST['submit']))
{
    $err = [];
    // проверяем логин
    if(!preg_match("/^[a-zA-Z0-9]+$/",$_POST['login']))
    {
        $err[] = "Логин может состоять только из букв английского алфавита и
цифр";
    }
    // проверяем, не существует ли пользователя с таким именем
    $query = mysqli_query($link, "SELECT user_id FROM users WHERE
user_login='".mysqli_real_escape_string($link, $_POST['login'])."'");
    if(mysqli_num_rows($query) > 0)
    {
        $err[] = "Пользователь с таким логином уже существует в базе данных";
    }

    // Если нет ошибок, то добавляем в БД нового пользователя
    if(count($err) == 0)
    {
        $login = $_POST['login'];
        // Убираем лишние пробелы и делаем двойное хеширование
        $password = md5(md5(trim($_POST['password'])));
```

```

        mysqli_query($link,"INSERT INTO users SET user_login='".$login."',
user_password='".$password.'");
        header("Location: check.php"); exit();
    }
    else
    {
        print "<b>При регистрации произошли следующие ошибки:</b><br>";
        foreach($err AS $error)
        {
            print $error."<br>";
        }
    }
}

```

Авторизация пользователя:

```

<?php
// Скрипт проверки
require_once 'connection.php';
$link=mysqli_connect($host, $user, $password, $database)or die("Ошибка " .
mysqli_error($link));
if (isset($_COOKIE['id']) and isset($_COOKIE['hash']))
{
    $query = mysqli_query($link, "SELECT * FROM users WHERE user_id =
''.intval($_COOKIE['id']).'' LIMIT 1");
    $userdata = mysqli_fetch_assoc($query);
    if(($userdata['user_hash'] !== $_COOKIE['hash']) or ($userdata['user_id'] !==
$_COOKIE['id']))
    {
        setcookie("id", "", time() - 3600*24*30*12, "/");
        setcookie("hash", "", time() - 3600*24*30*12, "/", null, null, true);
        print "Хм, что-то не получилось";
    }
    else {header ("Location: main_page.php"); exit();}
}
?>

```

Вывод каталога товаров:

```

<div class="row">
    <?php
        require_once
'D:/8_semestr/php/OSPanel/domains/localhost/connection.php';
        $link = mysqli_connect($host, $user, $password, $database)
            or die("Ошибка " . mysqli_error($link));

```

```

        $query = "SELECT * FROM items";
        $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
        if($result){
            $rows = mysqli_num_rows($result);
            for ($i = 0 ; $i < $rows ; ++$i){
                $row = mysqli_fetch_row($result);?>
<div class="gallery">
    <div class="product-img">
        <?php echo "<a href='#>$row[1]</a>";?>
    </div>
    <p class="product-title">
        <?php echo "<a href='#>$row[2]</a>";?>
    </p>
    <?php
        echo "<p class='desc'>$row[4]</p>";
        echo "<a href='add_to_basket.php?click=1'
class='button11'>$row[3] &#8381</a>";
    </div>
    <?php
        }
        mysqli_free_result($result);}
        mysqli_close($link);?>
</div>

```

Добавление в корзину:

```

function add_to_basket(){
    require_once 'connection.php';
    $link = mysqli_connect($host, $user, $password, $database) or
die("Ошибка " . mysqli_error($link));
    global $link;
    $count=1;
    $item_id = $_POST['item_id'];
    $query = "INSERT INTO basket(basket_item_name, basket_item_quantity,
basket_item_price) SELECT items.item_name, '{$count}', items.item_price
FROM items WHERE items.item_id='{$item_id}';";
    $result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
    mysqli_free_result($result);
}

```

Вывод корзины:

```

<?php
    require_once 'connection.php';

```

```

$link = mysqli_connect($host, $user, $password, $database) or die("Ошибка
" . mysqli_error($link));
$query = "SELECT `basket_item_name`, `basket_item_quantity`,
`basket_item_price` FROM `basket`";
$result = mysqli_query($link, $query) or die("Ошибка " .
mysqli_error($link));
if($result){
    $total_cost = 0;
    $rows = mysqli_num_rows($result);
    echo "<h4>Корзина <span class='price' style='color:black'><i
class='fa fa-shopping-cart'></i> <b>$rows</b></span></h4>";
    for ($i = 0 ; $i < $rows ; ++$i) {
        $row = mysqli_fetch_row($result);
        echo "<p>$row[0]<span class='price'>$row[1]x$row[2]<i
class='fa fa-rub' aria-hidden='true'></i></span></p>";
        $total_cost += $row[1]*$row[2];
    }
    echo "<hr><p>Всего к оплате <span class='price'
style='color:black'><b>$total_cost</b><i class='fa fa-rub' aria-
hidden='true'></i></span></p>";
    mysqli_free_result($result);
}
mysqli_close($link);
?>

```

ЛИТЕРАТУРА

1. Котляров В.П. Основы тестирования программного обеспечения [Электронный ресурс] / В.П. Котляров. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 334 с. — Режим доступа: <http://www.iprbookshop.ru/62820.html>
2. Влацкая И.В. Проектирование и реализация прикладного программного обеспечения [Электронный ресурс] : учебное пособие / И.В. Влацкая, Н.А. Заельская, Н.С. Надточий. — Оренбург: Оренбургский государственный университет, ЭБС АСВ, 2015. — 119 с. — Режим доступа: <http://www.iprbookshop.ru/54145.html>
3. Гагарин, Ю.Е. Модульное тестирование: Методические указания по выполнению домашнего задания по курсу «Тестирование и отладка программного обеспечения» / Ю.Е. Гагарин. — Калуга: КФ МГТУ им. Н.Э. Баумана, 2017. — 20 с.
4. Майерс, Г. Искусство тестирования программ / Г. Майерс. — М.: Вильямс, 2012. — 272 с.
5. Канер, С. Тестирование программного обеспечения / С. Канер. — Киев: ДиаСофт, 2001. — 544 с.
6. Калбертсон, Р. Быстрое тестирование / Р. Калбертсон, К. Браун, Г. Кобб. — М.: Вильямс, 2002. — 384 с.
7. Липаев В.В. Тестирование компонентов и комплексов программ. Учебник. — М.: СИНТЕГ, 2010. — 400 с.
8. Макгрегор, Д. Тестирование объектно-ориентированного программного обеспечения / Д. Макгрегор, Д. Сайкс. — Киев: ДиаСофт, 2002. — 348 с.
9. Бейзер, Б. Тестирование «черного ящика» / Б. Бейзер. — Киев: ДиаСофт, 2002. — 326 с.
10. Лоскутов, В.И. Разработка информационных систем для Windows Store [Электронный ресурс] / В.И. Лоскутов, И.Л. Коробова. — Москва : Национальный Открытый Университет «ИНТУИТ», 2016. — 180 с. — Режим доступа: <http://biblioclub.ru/index.php?page=book&id=428809>