УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по практическому заданию №1

по предмету «Метрология, стандартизация и сертификация (в информационных технологиях»

Тема работы: Метрики Холстеда

Выполнили:

Кисель П.А.,

Сологуб Е.М.

Гр. 351003

Проверила:

Болтак С. В.

Минск 2025

1. **Код php:**

<?php

class UserManager {

private $db;

private $users = [] ;

public function \_\_construct ( $database ) {

$this->db = $database ;

$this->loadUsers() ;

}

public function loadUsers() {

while ( $row = $result->fetch\_assoc() ) {

$this->users[ $row['id']] = $row;

}

}

public function getUserById($userId) {

return $this -> users[ $userId ] ?? null;

}

public function addUser( $name, $email, $role = 'user') {

if (!filter\_var( $email, FILTER\_VALIDATE\_EMAIL) || strlen( $name) < 2) {

throw new Exception( " Invalid input " ) ;

}

$stmt = $this->db->prepare( " INSERT INTO users (name, email, role) VALUES ( ? , ? , ? ) " );

$stmt->bind\_param( " sss " , $name, $email, $role) ;

if ( $stmt -> execute () ) {

$newUserId = $this-> db -> insert\_id;

$this->users[ $newUserId] = [' name ' => $name , ' email ' => $email, ' role ' => $role] ;

return $newUserId;

}

return false;

}

public function updateUser( $userId, $data) {

if ( ! isset( $this-> users [ $userId ] ) ) {

throw new Exception (" User not found ") ;

}

$allowed = [ ' name ', 'email', 'role' ] ;

$updateData = array\_intersect\_key( $data , array\_flip ( $allowed) ) ;

if ( empty ( $updateData ) ) return false;

$setClause = implode ( " , " , array\_map ( fn ( $f ) => " $f = ? ", array\_keys ( $updateData ) ) ) ;

$params = array\_merge ( array\_values ( $updateData ) , [ $userId] ) ;

$types = str\_repeat( "s" , count( $updateData) ) . " i ";

$stmt->bind\_param( $types, ... $params) ;

if ( $stmt->execute( ) ) {

$this->users[ $userId] = array\_merge( $this->users[ $userId] , $updateData) ;

return true ;

}

return false ;

}

public function deleteUser( $userId) {

if ( ! isset( $this->users[ $userId] ) ) return false ;

$stmt->bind\_param("i", $userId) ;

if ( $stmt->execute() ) {

unset( $this->users[ $userId] ) ;

return true;

}

return false;

}

public function getUsersByRole ( $role) {

return array\_filter( $this->users, fn( $user) => $user['role'] === $role) ;

}

}

$db = new mysqli( " localhost ", " user ", " pass ", " test\_db ") ;

$manager = new UserManager( $db) ;

try {

$id = $manager->addUser("John", "john@mail.com", "admin") ;

echo "User $id added\n" ;

$user = $manager->getUserById ( $id ) ;

if ( $user) echo "Found: {$user['name']}\n";

$manager->updateUser( $id, ['name' => ' John Updated '] ) ;

echo "User updated \n "

$admins = $manager->getUsersByRole('admin') ;

echo "Admins: " . count( $admins) . "\n";

} catch (Exception $e) {

echo "Error: " . $e->getMessage() . "\n";

}

?>

1. **Ручной расчёт:**

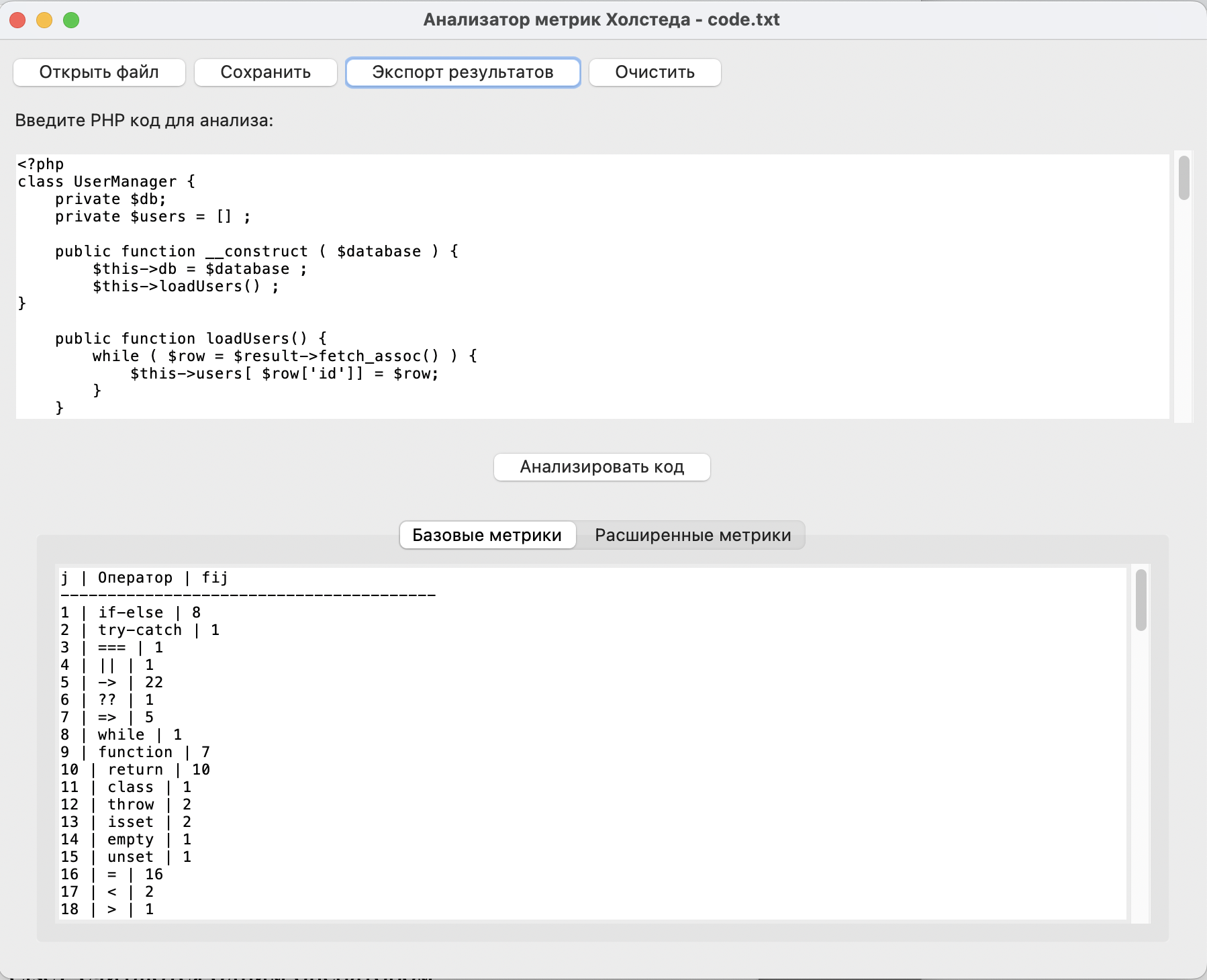
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***j*** | **Оператор** | ***f1j*** | ***i*** | **Операнд** | ***zf2i*** |
| 1 | class | 1 | 1 | $db | 3 |
| 2 | {} | 16 | 2 | $users | 1 |
| 3 | = | 16 | 3 | $database | 2 |
| 4 | () | 55 | 4 | $userId | 11 |
| 5 | -> | 22 | 5 | $name | 4 |
| 6 | ; | 32 | 6 | $email | 4 |
| 7 | If-else | 8 | 7 | $role | 5 |
| 8 | ! | 3 | 8 | $data | 2 |
| 9 | while | 1 | 9 | $allowed | 2 |
| 10 | \_\_construct | 1 | 10 | $updateData | 6 |
| 11 | return | 10 | 11 | $setClause | 1 |
| 12 | ?? | 1 | 12 | $params | 2 |
| 13 | empty | 1 | 13 | $types | 2 |
| 14 | . | 4 | 14 | $stmt | 7 |
| 15 | === | 1 | 15 | $row | 3 |
| 16 | function | 7 | 16 | $result | 1 |
| 17 | Try catch | 1 | 17 | $newUserId | 3 |
| 18 | [] | 15 | 18 | $user | 4 |
| 19 | => | 5 | 19 | $e | 2 |
| 20 | || | 1 | 20 | $manager | 5 |
| 21 | < | 2 | 21 | $id | 3 |
| 22 | implode | 1 | 22 | $admins | 2 |
| 23 | count | 2 | 23 | $f | 1 |
| 24 | execute | 3 | 24 | 2 | 1 |
| 25 | str\_repeat | 1 | 25 | true | 2 |
| 26 | strlen | 1 | 26 | false | 5 |
| 27 | array\_intersect\_key | 1 | 27 | name | 3 |
| 28 | array\_flip | 1 | 28 | email | 2 |
| 29 | array\_map | 1 | 29 | role | 3 |
| 30 | array\_keys | 1 | 30 | INSERT INTO users (name, email, role) VALUES (?, ?, ?) | 1 |
| 31 | array\_values | 1 | 31 | sss | 1 |
| 32 | array\_merge | 2 | 32 | User not found | 1 |
| 33 | array\_filter | 1 | 33 | John Updated | 1 |
| 34 | fetch\_assoc | 1 | 34 | i | 2 |
| 35 | getUserById | 2 | 35 | localhost | 1 |
| 36 | getMessage | 1 | 36 | pass | 1 |
| 37 | getUsersByRole | 2 | 37 | John | 1 |
| 38 | addUser | 2 | 38 | john@mail.com | 1 |
| 39 | loadUsers | 2 | 39 | User $id added | 1 |
| 40 | bind\_param | 3 | 40 | user | 2 |
| 41 | updateUser | 2 | 41 | User updated | 1 |  |
| 42 | mysqli | 2 | 42 | Admins | 1 |
| 43 | Exception | 4 | 43 | admin | 2 |
| 44 | UserManager | 2 | 44 | Error | 1 |
| 45 | > | 1 | 45 | $this | 13 |
| 46 | , | 23 | 46 | s | 1 |
| 47 | ？ | 2 | 47 | test\_db | 1 |
| 48 | isset | 2 | 48 | Found: {$user['name']} | 1 |
| 49 | unset | 1 | 49 | Invalid input | 1 |
| 50 | throw | 2 | 50 | id | 1 |
| 51 | filter\_var | 1 | 51 | , | 1 |
| 52 | deleteUser | ***1*** | 52 | $f = ? | 1 |
| 53 | prepare | ***1*** | 53 | null | 1 |
| **η1 *= 53*** |  | ***N1= 276*** | 54 | \n | 2 |
|  |  |  | **η2 *= 54*** |  | ***N2 = 133*** |
|  |  |  |  |  |  |

Словарь программы **η**=53 + 54 = 107.

Длина программы ***N*** = 276 + 133= 409.

Объем программы ***V*** = 409 \* = 2757.25999744

1. **Результат работы программы**



Экспорт в CSV-формат:

