**Організація тестування**

Для тестування використовувалося середовище розробки IntelliJ IDEA Ultimate та фреймворк автоматичного тестування Junit 5. Написання тестів велося на основі проекту тестових наборів, описаних в лабораторній роботі №8.

Створення тестів велося у спеціальній директорії проекту з назвами, що відображають клас, що тестується типу OrderTest. Кожен окремий набір тестових даних виділявся анотацією @Test та мав включати очікувані та реальні дані, на основі порівняння яких вирішувалося, чи був тест пройдений.

**Тестування**

Функції для тестування обиралися виходячи з найбільшої вірогідності виникнення в них помилки.

Специфікація функції getDishNames класу Order має наступний опис.

Вхідні параметри відсутні.

Вихідні дані: текстовий рядок, що включає усі назви страв та їх кількість для заданого замовлення у вигляді «Кава (2 од.), Тістечко (1 од.)»

Таблиця 1 – Опис тестових наборів для функції getDishNames

|  |  |  |
| --- | --- | --- |
| **№ тесту** | **Опис значень початкових даних** | **Опис очікуваних значень результату** |
| 1 | *Поточні дані - details = {[id\_details=1, id\_order=1,* ***id\_dish=1****,* ***quantity=2****,\*order,\*dish[“dish1”]]; [id\_details=1, id\_order=1,* ***id\_dish=2****,* ***quantity=1****,\*order,\*dish[“dish2”]]; [id\_details=1, id\_order=1,* ***id\_dish=3****,* ***quantity=2****,\*order,\*dish[“dish2”]]}* | dish1 (2 шт.), dish2 (1 шт.), dish3 (2шт.) |

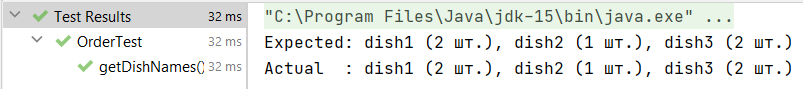


Рис. 2 – результат тесту 1

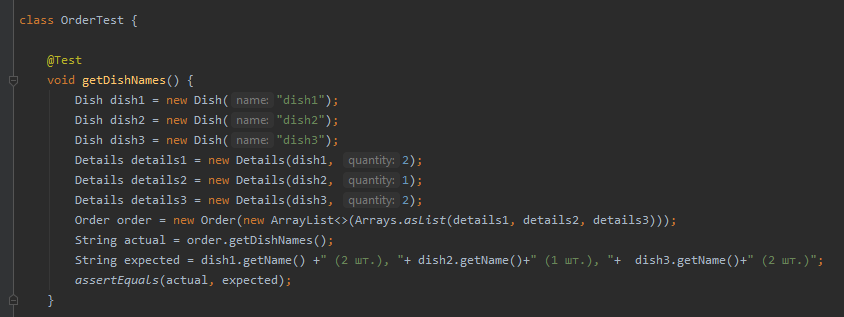


Рис. – програмний код тесту 1

Специфікація функції addDetail класу Order має наступний опис.

Вхідні параметри:

1. detail – дані замовленої страви (Об’єкт класу Details):

* містить посилання на об’єкт страви та її замовлену кількість, також інші поля (власне id, id замовленої страви та id замовлення, , посилання на об’єкт замовлення), які не задіяні в тестуванні або не впливають на очікуваний результат чи генеруються автоматично.

Значення не повертаються функцією. Для перевірки коректності роботи, викликається функція getDishNames, що повертає текстову строку, що включає усі назви страв та їх кількість для заданого замовлення у вигляді «Кава (2 од.), Тістечко (1 од.)»

Таблиця 2 – Опис тестових наборів для функції addDetail

|  |  |  |
| --- | --- | --- |
| **№ тесту** | **Опис значень початкових даних** | **Опис очікуваних значень результату** |
| 2 | *Поточні дані -* dish1 (2 шт.), dish2 (1 шт.), dish3 (2шт.)  *Тобто details = {[ quantity=2, \*dish[“dish1”]]; [ quantity=1, \*dish[“dish2”]]; [ quantity=2, \*dish[“dish3”]]}*  Вхідні дані:  detail: {quantity=3, \*[“dish4”]} | dish1 (2 шт.), dish2 (1 шт.), dish3 (2шт.) |

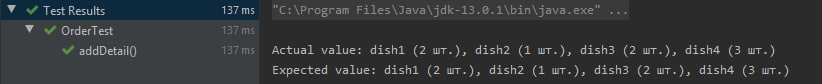
Специфікація функції getDetailsIfPresent класу Order має наступний опис.

Рис. 4 – результат тесту 2

Рис. 3 – програмний код тесту 2

Вхідні параметри:

1. id – ідентифікатор страви, що запитується (ціле число).

Значення, що повертаються функцією:

detail – об’єкт однойменного класу, що містить деталі замовленої страви;

null – якщо об’єкт з даним id не знайдено

|  |  |  |
| --- | --- | --- |
| **№ тесту** | **Опис значень початкових даних** | **Опис очікуваних значень результату** |
| 4 | *Поточні дані -*  *details = {[id\_details=1, id\_order=1,* ***id\_dish****=1, quantity=2, \*order, \*dish[“dish1”]]; [id\_details=1, id\_order=1,* ***id\_dish=2****, quantity=1,\*order,\*dish[“dish2”]];}*  Вхідні дані: id = 1 | details[dish1 (2 шт.)] |
| 5 | *Поточні дані - details = {[id\_details=1, id\_order=1,* ***id\_dish****=1, quantity=2, \*order, \*dish[“dish1”]]; [id\_details=1, id\_order=1,* ***id\_dish=2****, quantity=1,\*order,\*dish[“dish2”]];}*  Вхідні дані: id = 100 | null |

Таблиця 3 – Опис тестових наборів для функції getDetailsIfPresent

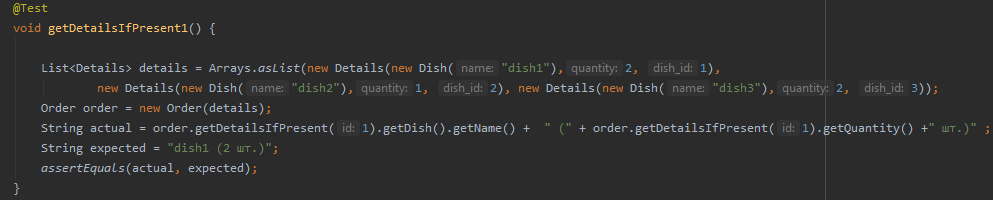


Рис. 5 – програмний код тесту 4

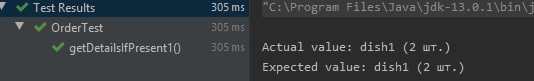
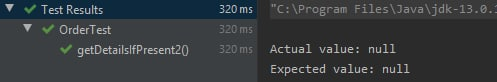


Рис. 8 – результат тесту 5

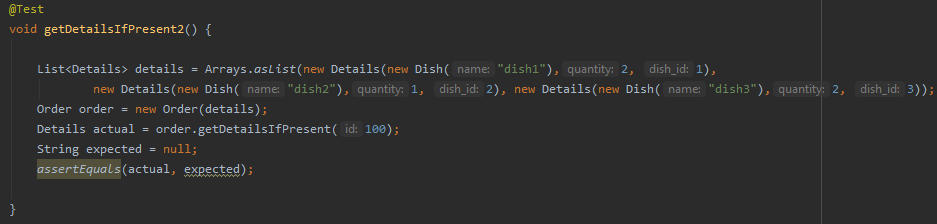


Рис. 7 – програмний код тесту 5

Рис. 6 – результат тесту 4

Специфікація функції sortByQuantity класу Order має наступний опис.

Вхідні параметри відсутні.

Значення не повертаються функцією. Для перевірки коректності роботи, викликається функція getDishNames.

|  |  |  |
| --- | --- | --- |
| **№ тесту** | **Опис значень початкових даних** | **Опис очікуваних значень результату** |
| 6 | *Поточні дані -*  Випадкова кількість замовлених страв від 1 до Х  Наприклад: Кава (2 од.) , Тістечко (1 од.), Печиво (3 од.) | *Відсортовані елементи за спадом*  Наприклад: Кава (2 од.) , Тістечко (1 од.), Печиво (3 од.) |

Таблиця 4 – Опис тестових наборів для функції sortByQuantity

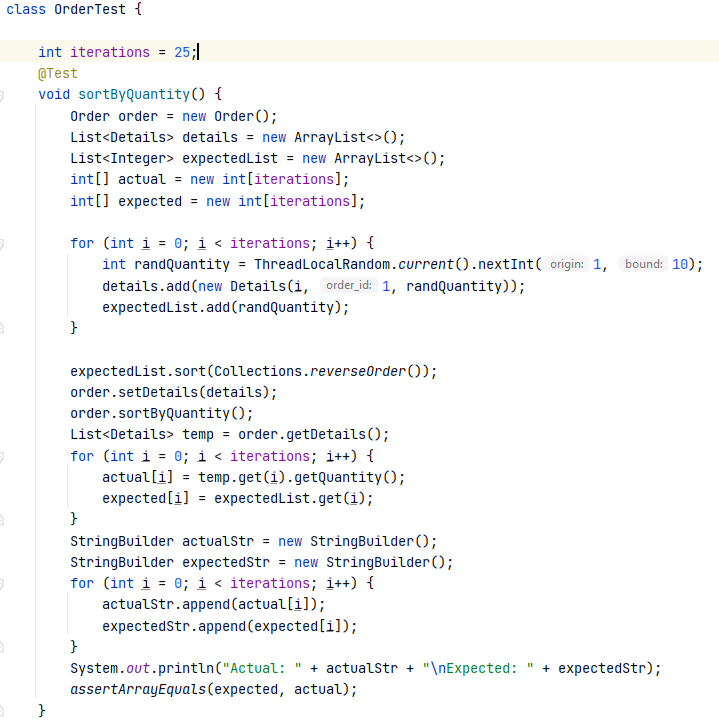


Рис. 9 – програмний код тесту 6

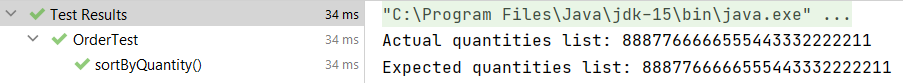


Рис. 10 – результат тесту 6

Специфікація функції getCost класу Details має наступний опис.

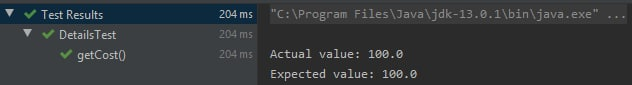
Вхідні параметри відсутні.

Значення, що повертаються функцією:

число з плаваючою комою – добуток кількості замовлених страв на їх вартість;

Таблиця 5 – Опис тестових наборів для функції getCost класу Details

|  |  |  |
| --- | --- | --- |
| **№ тесту** | **Опис значень початкових даних** | **Опис очікуваних значень результату** |
| 7 | *Поточні дані -*  *details = {quantity = 2, \*dish [name = “dish1”,*  *price = 50]]}* | 100.0 |

Специфікація функції getCost класу Order має наступний опис.

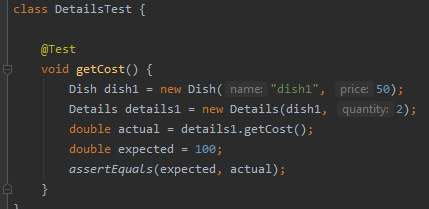


Рис. 11 – програмний код тесту 7

Рис. 12 – результат тесту 7

Вхідні параметри відсутні.

Значення, що повертаються функцією:

число з плаваючою комою – сума добутків усієї кількості замовлених страв на їх вартості;

|  |  |  |
| --- | --- | --- |
| **№ тесту** | **Опис значень початкових даних** | **Опис очікуваних значень результату** |
| 8 | *Поточні дані - details =*  *{[****quantity=2****, \*dish [name = “dish1”,* ***price = 50****]]; [****quantity=1****, \*dish [name = “dish2”,* ***price = 100****]]; [****quantity=2****, \*dish [name = “dish3”,* ***price = 10****]]}* | 220.0 |

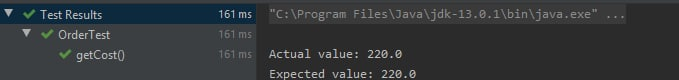
Таблиця 6 – Опис тестових наборів для функції getCost класу Order

Рис. 3 – результат тесту 8

Рис. – програмний код тесту 8

Специфікація функції removeFromOrder класу Order має наступний опис.

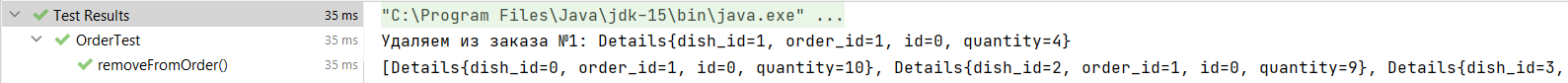
Вхідні параметри:

id – ідентифікатор страви, що має бути видаленою з замовлення (ціле число)

Значення, що повертаються функцією відстуні, окрім випадку відсутності видаляємого елемента – в такому разі виникає виняток - NoSuchElementException. Для перевірки коректності роботи, викликається функція getDishNames.

Таблиця 7 – Опис тестових наборів для функції removeFromOrder класу Order

|  |  |  |
| --- | --- | --- |
| **№ тесту** | **Опис значень вхідних даних** | **Опис очікуваних значень результату** |
| 9 | *Випадкові дані - наприклад*  *Кава (2 од.) [id=1], Тістечко (1 од.) [id=2], Печиво (3 од.) [id=3]*  id=2 | *Кава (2 од.), Печиво (3 од.)* |
| 10 | *Випадкові дані - наприклад*  *Кава (2 од.) [id=1], Печиво (3 од.) [id=3]*  id=2 | *NoSuchElementException* |



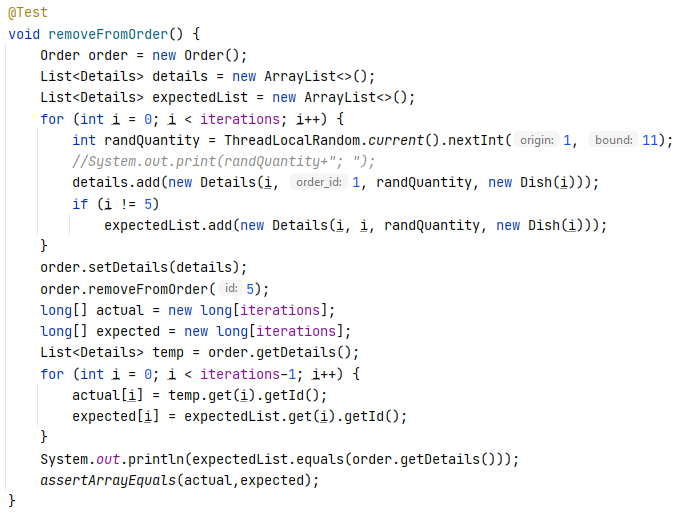


Рис. 4 – програмний код тесту 9

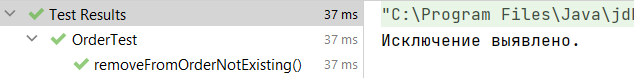


Рис. 17 – результат тесту 10

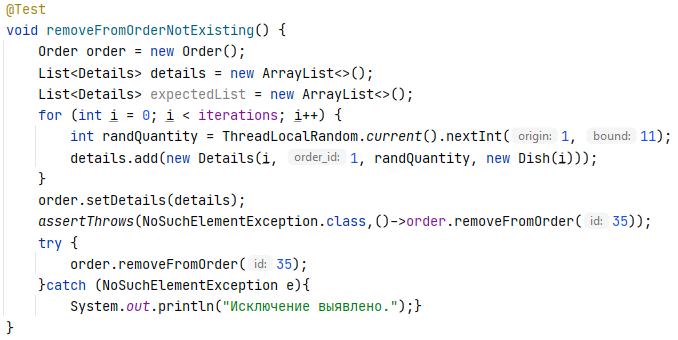


Рис. 6 – програмний код тесту 10

Рис. 15 – результат тесту 9