

Аннотация теста

Название проекта	PM02_4ISIP520_Rekhviashvili_1
Рабочая версия	v 0.1
Имя тестирующего	Владимир
Дата(ы) теста	17.04.2024г

Расшифровка тестовых информационных полей:

Поле	Описание
Название проекта	Название тестируемого проекта
Рабочая версия	Версия проекта/программного обеспечения (первый тест считается 1.0).
Имя тестирующего	Имя того, кто проводил тесты
Дата(ы) теста	Дата(ы) проведения тестов – это один или несколько дней. Если тесты проводились в более протяженный период времени, нужно отметить отдельную дату для каждого теста.
Тестовый пример #	Уникальный ID для каждого тестового примера. Следуйте некоторым конвенциям, чтобы указать типы тестов. Например, 'ТС_UI_1' означает 'user interface test case #1' (ТС_ПИ_1: тестовый случай пользовательского интерфейса#1)
Приоритет тестирования (Низкий/Средний/Высокий)	Насколько важен каждый тест. Приоритет тестирования для бизнес-правил и функциональных тестовых случаев может быть средним или высоким, в то время как незначительные случаи пользовательского интерфейса могут иметь низкий приоритет.
Заголовок/название теста	Название тестового случая. Например, Подтвердите страницу авторизации с действительным именем пользователя и паролем.
Краткое изложение теста	Описание того, что должен достичь тест.
Этапы теста	Перечислите все этапы теста подробно. Запишите этапы теста в том порядке, в котором они должны быть реализованы. Предоставьте как

	можно больше подробностей и разъяснений. Пронумерованный список – хорошая идея.
<b>Тестовые данные</b>	Перечислите/опишите все тестовые данные, используемые для данного тестового случая. Так, фактические используемые входные данные можно отслеживать по результатам тестирования. Например, Имя пользователя и пароль для подтверждения входа.
<b>Ожидаемый результат</b>	Каким должен быть вывод системы после выполнения теста? Подробно опишите ожидаемый результат, включая все сообщения/ошибки, которые должны отображаться на экране.
<b>Фактический результат</b>	Каким должен быть фактический результат после выполнения теста? Опишите любое релевантное поведение системы после выполнения теста.
<b>Предварительное условие</b>	Любые предварительные условия, которые должны быть выполнены до выполнения теста. Перечислите все предварительные условия для выполнения этого тестового случая.
<b>Постусловие</b>	Каким должно быть состояние системы после выполнения теста?
<b>Статус</b> <i>(Зачет/Незачет)</i>	Если фактический результат не соответствует ожидаемому результату, отметьте тест как неудачный. В ином случае обновление пройдено.
<b>Примечания/комментарии</b>	Используйте эту область для любых дополнительных замечок/комментариев/вопросов. Эта область предназначена для поддержки вышеуказанных полей (например, если есть некоторые особые условия, которые не могут быть описаны в любом из вышеуказанных полей, или если есть вопросы, связанные с ожидаемыми или фактическими результатами).

Тестовый пример #1:

Тестовый пример #	ТС_1
Приоритет тестирования	Высокий
Заголовок/название теста	Проверка на результат вывод
Краткое изложение теста	Проверка вывода после ввода данных
Этапы теста	1.Ввод данных 2. Расчёт данных с помощью функций. 3. Вывод ответа в консольную строку.
Тестовые данные	Количество поставщиков = 5 Количество потребителей = 3 Запасы поставщиков = 20, 45, 24, 31, 30 Нужды потребителей = 65, 44, 41 Стоимости перевозок = 5, 4, 6; 7, 3, 3; 9, 5, 2; 3, 2, 5; 4, 7, 1.
Ожидаемый результат	Опорный план перевозок = 594
Фактический результат	См. рисунок 1
Статус	Зачёт
Предварительное условие	Сумма запасов всех поставщиков = сумма нужд всех потребителей
Постусловие	Вывод без ошибок.
Примечания/комментарии	-

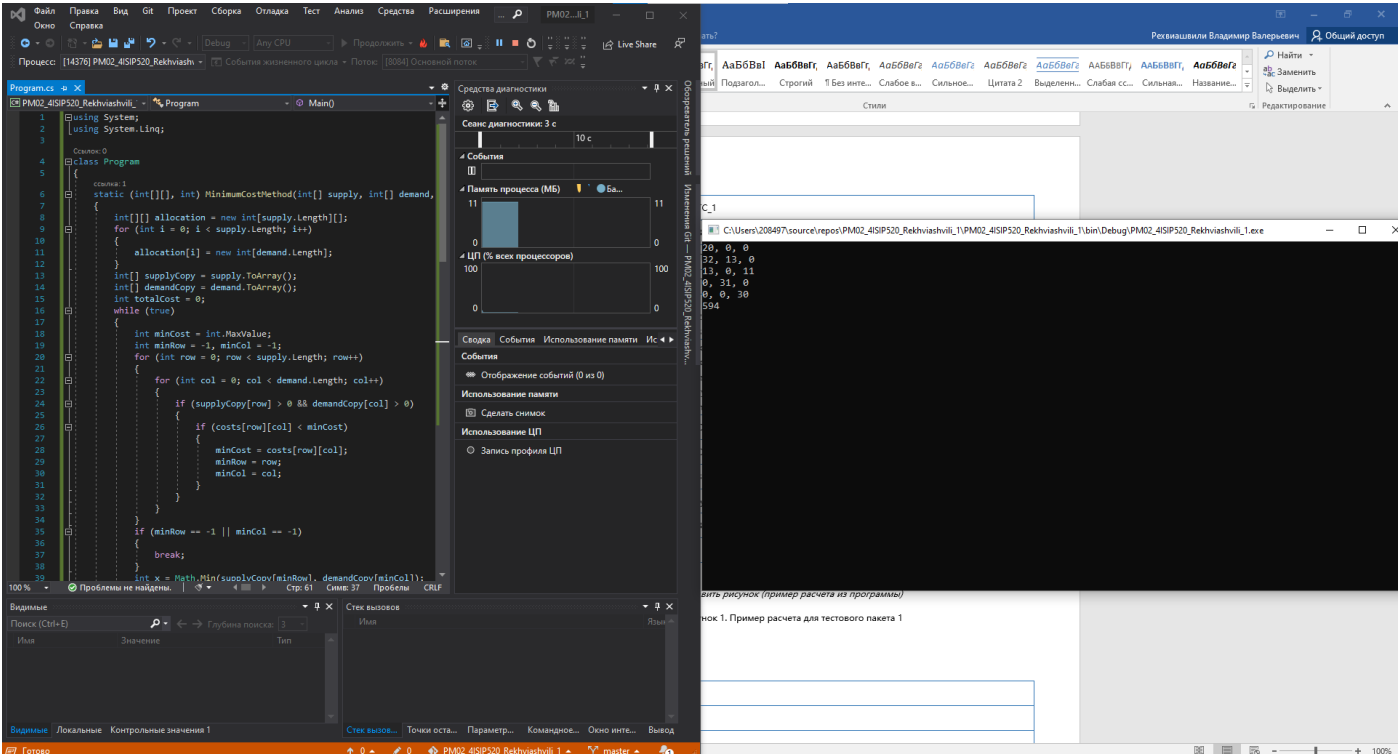


Рисунок 1.

Тестовый пример #2:

Тестовый пример #	ТС_2
Приоритет тестирования	Высокий
Заголовок/название теста	Проверка расчёта пути
Краткое изложение теста	Проверка вывода после ввода данных
Этапы теста	1.Ввод данных 2. Расчёт данных с помощью функций. 3. Вывод ответа в консольную строку.
Тестовые данные	Количество поставщиков = 5 Количество потребителей = 3 Запасы поставщиков = 20, 45, 24, 31, 30 Нужды потребителей = 65, 44, 41 Стоимости перевозок = 5, 4, 6; 7, 3, 3; 9, 5, 2; 3, 2, 5; 4, 7, 1.
Ожидаемый результат	Опорный план перевозок = 594
Фактический результат	См. рисунок 2
Статус	Зачёт
Предварительное условие	Сумма запасов всех поставщиков = сумма нужд всех потребителей
Постусловие	Вывод без ошибок.
Примечания/комментарии	-

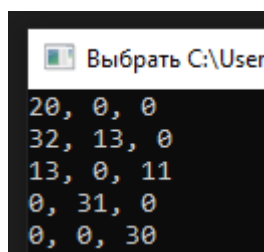


Рисунок 2.

Тестовый пример #3:

Тестовый пример #	ТС_3
Приоритет тестирования	Средний
Заголовок/название теста	Проверка на оптимизацию
Краткое изложение теста	В случае, если задача не оптимизирована, вывод в консоли соответствующего сообщения
Этапы теста	1.Ввод данных 2. Расчёт данных с помощью функций.

	3. Вывод ответа в консольную строку
Тестовые данные	Количество поставщиков = 5 Количество потребителей = 3 Запасы поставщиков = 20, 45, 24, 31 Нужды потребителей = 65, 44, 41 Стоимости перевозок = 5, 4, 6; 7, 3, 3; 9, 5, 2; 3, 2, 5; 4, 7, 1.
Ожидаемый результат	Вывод сообщения «Задача не оптимальна»
Фактический результат	См. рисунок 3
Статус	Зачёт
Предварительное условие	Сумма запасов всех поставщиков != сумма нужд всех потребителей
Постусловие	Вывод с предупреждением
Примечания/комментарии	-

7  
8  
9  
0  
1  
2  
3  
4  
5  
6  
7  
8  
9

Ссылка: 0

```
public static void Main()
{
    int sum1 = 0;
    int sum2 = 0;
    int[] supply = { 20, 45, 24, 31};
    int[] demand = { 65, 44, 41 };
    int[][] costs = new int[][]
    {
        new int[] { 5, 4, 6 },
        new int[] { 7, 3, 3},
        new int[] { 9, 5, 2 },
        new int[] { 3, 2, 5 },
        new int[] { 4, 7, 1 }
    }
}
```

C:\Users\208497\source\repos\PM02\_4ISIP520\_Rekhviashvili\_1\PM02\_4ISIP520\_

Задача не оптимальна!!

Рисунок 3.