Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

ПНИПУ

**Калькулятор множеств**

**«»**

Выполнил:   
студент группы РИС-23-1б   
Молодых Никита Андрееевич

Проверила:   
доцент кафедры ИТАС   
О.Л.Викентьева

Пермь 2024 г.

**Постановка задачи:**

**Цель:**

Разработать калькулятор множеств, который выполняет следующие действия над множествами целых чисел:

* Позволяет задать универсальное множество U.
* Позволяет задать 5 множеств (A, B, C, D, E), элементы которых принадлежат U.
* Позволяет выбрать множества и выполнить операции над множествами A, B, C, D, E:

1. Пересечение
2. Объединение.
3. Разность.
4. Симметрическая разность.
5. Дополнение.

* Позволяет проверить принадлежит ли заданный элемент множеству (элемент вводится с клавиатуры).

Позволяет проверить входит ли одно подмножество в другое.

**Необходимо предусмотреть:**

* Проверку правильности вводимых элементов (например, выводится ошибка, если пользователь вводит дробное число вместо целого)
* Два способа формирования множеств (ввод с клавиатуры и генерацию датчиком случайных чисел).
* Интерфейс с пользователем в виде текстового меню.
* В множество элемент должен включаться только один раз (т.е. элементы множества не дублируются).

**Результаты:**

* Вывести результаты всех вычислений на экран.
* Объяснить результаты работы программы и провести анализ тестов.

**Методы:**

* Использовать вспомогательные переменные для хранения промежуточных значений.
* Выполнить приведение типов данных при работе с вещественными числами.
* Проанализировать достаточность тестов по критериям черного и белого ящика.

1. Пересечение (A ∩ B):

- Это множество, содержащее элементы, которые присутствуют одновременно в обоих множествах A и B.

- Пример: Если A = {1, 2, 3} и B = {2, 3, 4}, то A ∩ B = {2, 3}.

2. Объединение (A ∪ B):

- Это множество, содержащее все элементы, которые есть хотя бы в одном из множеств A или B.

- Пример: Если A = {1, 2, 3} и B = {2, 3, 4}, то A ∪ B = {1, 2, 3, 4}.

3. Разность (A \ B):

- Это множество, содержащее элементы, которые есть в множестве A, но отсутствуют в множестве B.

- Пример: Если A = {1, 2, 3} и B = {2, 3, 4}, то A \ B = {1}.

4. Симметрическая разность (A Δ B):

- Это множество, содержащее элементы, которые есть либо в A, либо в B, но не в обоих.

- Пример: Если A = {1, 2, 3} и B = {2, 3, 4}, то A Δ B = {1, 4}.

5. Дополнение (¬A):

- Это множество, содержащее все элементы, которые не входят в множество A, относительно некоторой универсальной множества U.

- Пример: Если U = {1, 2, 3, 4, 5} и A = {2, 3}, то ¬A = {1, 4, 5}.

Каждая из этих операций имеет свои уникальные свойства и применяется в различных областях математики и информатики. Если у вас есть дополнительные вопросы по этой теме, не стесняйтесь задавать!

**Тесты:**

**1)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Switch | Тест | Ожидаемый результат | полученный результат |
| 1 | A | {1,2,3,4,5} | {1,2,3,4,5} |  |
| 2 | B | {1,2,3,4,5} | {1,2,3,4,5} |  |
| 3 | C | {1,2,3,4,5} | {1,2,3,4,5} | + |
| 4 | D | {1,2,3,4,5} | {1,2,3,4,5} |  |
| 5 | E | {1,2,3,4,5} | {1,2,3,4,5} |  |
| 6 | A | Random | Random |  |
| 7 | B | Random | Random |  |
| 8 | C | Random | Random | + |
| 9 | D | Random | Random |  |
| 10 | E | Random | Random |  |
| 11 | A ∪ B | A={1,2,3,4,5};B={4,5,6,7,8} | 2 3 4 5 6 7 8 | + |
| 12 | A ∪ C | A={1,2,3,4,5};C={8,9,10,11} | 2 3 4 5 6 8 9 10 11 | + |
| 13 | A ∪ D | A={1,2,3,4,5};D={11,12,13} | 2 3 4 5 6 11 12 13 | + |
| 14 | A ∪ E | A={1,2,3,4,5};E={1,9,6,3,4} | 2 3 4 5 6 8 9 | + |
| 15 | A ∩ B | A={1,2,3,4,5};B={4,5,6,7,8} | 5 6 | + |
| 16 | A ∩ C | A={1,2,3,4,5};C={8,9,10,11} | Пустое множество | + |
| 17 | A ∩ D | A={1,2,3,4,5};D={11,12,13} | Пустое множество | + |
| 18 | A ∩ E | A={1,2,3,4,5};E={1,9,6,3,4} | 2 3 6 | + |
| 19 | A \ B | A={1,2,3,4,5};B={4,5,6,7,8} | 2 3 4 | + |
| 20 | A \ C | A={1,2,3,4,5};C={8,9,10,11} | 2 3 4 5 6 | + |
| 21 | A \ D | A={1,2,3,4,5};D={11,12,13} | 2 3 4 5 6 | + |
| 22 | A \ E | A={1,2,3,4,5};E={1,9,6,3,4} | 4 5 | + |
| 23 | A Δ B | A={1,2,3,4,5};B={4,5,6,7,8} | 2 3 6 7 8 | + |
| 24 | A Δ C | A={1,2,3,4,5};C={8,9,10,11} | 2 3 4 5 8 9 10 11 | + |
| 25 | A Δ D | A={1,2,3,4,5};D={11,12,13} | 2 3 4 5 11 12 13 | + |
| 26 | A Δ E | A={1,2,3,4,5};E={1,9,6,3,4} | 2 5 6 9 | + |
| 27 | ¬A | A={1,2,3,4,5};n=2 | Содержит | + |
| 28 | ¬B | B={4,5,6,7,8};n=7 | Содержит | + |
| 29 | ¬C | C={8,9,10,11};n=10 | Содержит | + |
| 30 | ¬D | D={11,12,13};n=11 | Содержит | + |
| 31 | ¬E | E={1,9,6,3,4};n=1 | Содержит | + |
| 32 | ¬A | A={1,2,3,4,5};n=10 | Не содержит | + |
| 33 | ¬B | B={4,5,6,7,8};n=10 | Не содержит | + |
| 34 | ¬C | C={8,9,10,11};n=1 | Не содержит | + |
| 35 | ¬D | D={11,12,13};n=11 | Не содержит | + |
| 36 | ¬E | E={1,9,6,3,4};n=2 | Не содержит | + |
| 37 | A | A={a,b,x} | Ошибка ввода | + |
| 38 | B | B={r,w,d} | Ошибка ввода | + |
| 39 | C | C={r,t,r} | Ошибка ввода | + |
| 40 | D | D={d,f,g} | Ошибка ввода | + |
| 41 | E | E={e,r,t} | Ошибка ввода | + |
| 42 | A | A={2.13} | Ошибка ввода | + |
| 43 | B | B={7.14} | Ошибка ввода | + |
| 44 | C | C={8.34} | Ошибка ввода | + |
| 45 | D | D={9.05} | Ошибка ввода | + |
| 46 | E | E={10.11} | Ошибка ввода | + |

**Анализ достаточности тестов:  
Черный ящик:**

1)



**Объяснение результатов:**

Эта программа призвана продемонстрировать следующие важные аспекты программирования:

1. Порядок выполнения операций. Она показывает, как последовательность операций может влиять на конечный результат.
2. Условные проверки. Программа наглядно иллюстрирует использование условных конструкций и их влияние на вычисления.
3. Влияние типов данных на точность. Она позволяет увидеть, как различные типы данных (например, целые числа и с плавающей запятой) могут влиять на точность вычислений.
4. Обработка ошибок. Программа демонстрирует подходы к контролю и обработке ошибок, возникающих в ходе вычислений.
5. Анализ результатов. Она дает возможность исследовать и анализировать результаты математических и логических операций.

Таким образом, данная программа является ценным инструментом для изучения и практики основополагающих концепций программирования.