

|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **"МИРЭА** - **Российский технологический университет"**  **РТУ МИРЭА** |

**Институт** Информационных Технологий

**Кафедра** математического обеспечения и стандартизации ИТ

**ПРАКТИЧЕСКАЯ РАБОТА №2**

**по дисциплине**

«Тестирование и верификация программного обеспеченья»

Студенты группы: ИКБО-05-20 Мочалов А.Д. *(Фамилия студента)*

Руководитель работы: Овчинникова М. А.

*(Фамилия преподавателя)*

|  |  |  |
| --- | --- | --- |
|  |  |  |

Москва 2022

1. **ДОКУМЕНТАЦИЯ ПО ПРОЕКТУ ФИО**

## Описание

Программа представляет собой оконное приложение, содержащие набор функций, позволяющий отсортировать введенный пользователем массив чисел использую алгоритм быстрой сортировки или алгоритм сортировки слиянием.

## Эксплуатация программы

Для запуска программы нужно извлечь файлы из архива и запустить исполняемый файл main.exe, после чего вы увидите главное окно программы (рисунок 1).

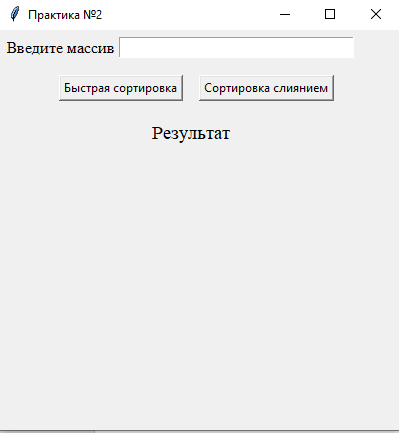
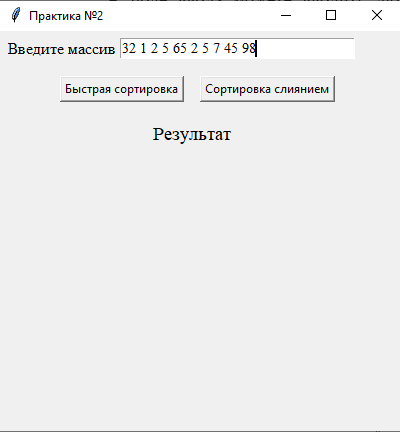


Рисунок 1 – Запуск программы

В поле ввода можете вводить любой порядок чисел через пробел, например, 32 2 1 5 65 2 5 7 45 98, после чего нажать на любую кнопку для применения сортировки (рисунок 2).



Риснуок 2 – Ввод значений

После нажатия на кнопку Быстрая сортировка появится результат сортировки (рисунок 3).

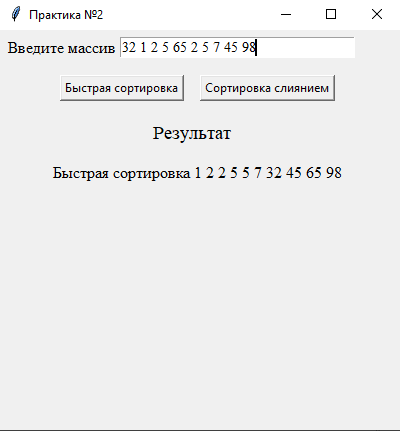
****

Рисунок 3 – Результат быстрой сортировки

После нажатия на кнопку «Сортировка слиянием» также появится результат сортировки, для этого вводить заново массив или перезапускать программу ненужно (рисунок 4).

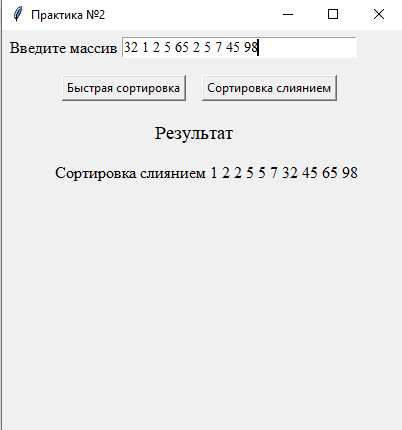


Рисунок 4 – Результат сортировки слиянием

1. **ПРОЕКТ ФИО**

## Тесты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Входные значения | Ожидаемый результат | Полученный результат | Комментарий |
| 1 | 7 | 13 | 13 | Программа соответствует документации продукта |
| 2 | 4 | 3 | 3 | Программа соответствует документации продукта |
| 3 | 12 | 144 | 144 | Программа соответствует документации продукта |
| 4 | 20 | 6765 | 6765 | Программа соответствует документации продукта |
| 5 | 7 | 5040 | 5040 | Программа соответствует документации продукта |
| 6 | 4 | 24 | 24 | Программа соответствует документации продукта |
| 7 | 12 | 479001600 | 479001600 | Программа соответствует документации продукта |
| 8 | 20 | 2432902008176640000 | 2432902008176640000 | Программа соответствует документации продукта |

## Замечания

Не обнаружены.

## Оценки

Функция поиска чисел Фибоначчи и поиска факториала числа работают корректно. Документация к проекту является исчерпывающей, позволяет в полной мере использовать программный продукт.

**ВЫВОД**

В результате выполнения практической работы был разработан программный продукт, написана документация, описывающая правила эксплуатации приложения. Были реализованы юнит тесты для чужого программного продукта, проведено тестирование, которое не выявило недочетов предоставленной программы.

1. **ДОКУМЕНТАЦИЯ ПО ПРОЕКТУ ФИО**
   1. **Описание**

Программа представляет собой консольное приложение, содержащие набор функций, позволяющий вычислять факториал числа, ряд Фибоначчи до числа, введенного пользователем, и число Фибоначчи согласно пользовательскому вводу. В программе реализован механизм мемоизации, что позволяет значительно ускорить вычисления.

* 1. **Эксплуатация программы**

От пользователя ожидается одно целое число на вход (Рисунок 5).



Рисунок 5 – Пользовательский ввод

После чего будет рассчитаны значения согласно описанию программы (Рисунок 6).

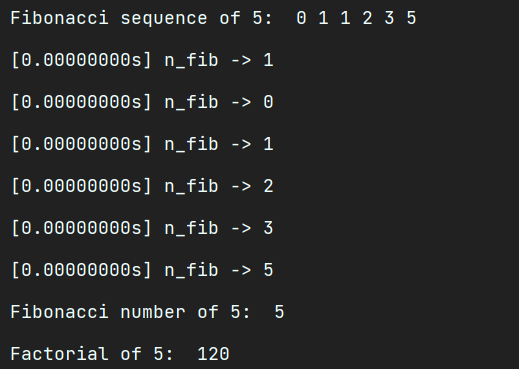


Рисунок 6 – Результат выполнения программы

1. **ПРОЕКТ ФИО**
   1. **Тесты**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название метода | Входные значения | Ожидаемый результат | Полученные значения | Комментарий |
| Find\_X | 0,0 | ALL | ALL | Метод работает корректно |
| Find\_x | 0,1 | NONE | NONE | Метод работает корректно |
| Find\_x | 5,6 | -1.2 | -1.2 | Метод работает корректно |
| Check\_palindrome | man | False | False | Метод работает корректно |
| Check\_palindrome | mom | True | True | Метод работает корректно |

* 1. **Замечания**

Не обнаружены.

* 1. **Оценки**

Алгоритмы сортировки работают в соответствии с целевой асимптотической сложностью. Документация к проекту является исчерпывающей, позволяет в полной мере использовать программный продукт.

**ВЫВОД**

В результате выполнения практической работы был разработан программный продукт, написана документация, описывающая правила эксплуатации приложения. Были реализованы юнит тесты для чужого программного продукта, проведено тестирование, которое не выявило недочетов предоставленной программы.

1. **ДОКУМЕНТАЦИЯ ПО ПРОЕКТУ ФИО**
   1. **Описание**

Программа содержит 2 функции, первая позволяет находить корни квадратных уравнений, а вторая проверяет, является ли строка палиндромом..

* 1. **Эксплуатация программы**

Пользователь должен ввести входные значения a, b квадратного уравнения и входную последовательность палиндромов (Рисунок 7).



Рисунок 7 – Пользовательский ввод

После чего будет рассчитаны значения согласно описанию программы (Рисунок 8).

****

Рисунок 8 – Результат выполнения программы

1. **ПРОЕКТ ФИО**
   1. **Тесты**

*Таблица Check*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название метода | Входные значения | Ожидаемый результат | Полученные значения | Комментарий |
| q\_sort | -1,2-10,2,4 | -10,-1,2,2,4 | -10,-1,2,2,4 | Метод работает корректно |
| q\_sort | 1,1,1,1,1,1 | 1,1,1,1,1,1 | 1,1,1,1,1,1 | Метод работает корректно |
| merge\_sort | -1,0,10,23.452 | -1,0,10,23.452 | -1,0,10,23.452 | Метод работает корректно |
| merge\_sort | -1,0,-32.34,6,4 | -32.34,-1,0,6,4 | -32.34,-1,0,6,4 | Метод работает корректно |

* 1. **Замечания**

Не обнаружены.

* 1. **Оценки**

Алгоритмы работают в соответствии с целевой асимптотической сложностью. Модульные тесты программы дают желаемые результаты.

**ВЫВОД**

В результате выполнения практической работы был разработан программный продукт, написана документация, описывающая правила эксплуатации приложения. Были реализованы юнит тесты для чужого программного продукта, проведено тестирование, которое не выявило недочетов предоставленной программы.

**ПРИЛОЖЕНИЯ**

Приложение А – Листинг программы тестирования

Приложение Б – Листинг программы тестирования

Приложение В – Листинг программы тестирования

**Приложение А**

Листинг программы тестирования

*Листинг А – Программа тестирования*

|  |
| --- |
| from main import Calculator  import unittest  class TestCalculator(unittest.TestCase):  def setUp(self):  self.calculator = Calculator()  def test\_fib(self):  self.assertEqual(self.calculator.n\_fib(7), 13)  self.assertEqual(self.calculator.n\_fib(4), 3)  self.assertEqual(self.calculator.n\_fib(12), 144)  self.assertEqual(self.calculator.n\_fib(20), 6765)  def test\_fact(self):  self.assertEqual(self.calculator.fact(7), 5040)  self.assertEqual(self.calculator.fact(4), 24)  self.assertEqual(self.calculator.fact(12), 479001600)  self.assertEqual(self.calculator.fact(20), 2432902008176640000)  if \_\_name\_\_ == "\_\_main\_\_":  unittest.main() |

**Приложение Б**

Листинг программы тестирования

*Листинг Б – Программа тестирования*

|  |
| --- |
| import unittest  from sort import \*  import random  class SortTest(unittest.TestCase):  def setUp(self) -> None:  pass  def test\_q\_sort(self):  self.assertEqual(q\_sort([234, 234, 55, 887.456, 11]), sorted([234, 234, 55, 887.456, 11]))  x = [random.random() for \_ in range(10)]  self.assertEqual(q\_sort(x), sorted(x))  def test\_merge\_sort(self):  self.assertEqual(merge\_sort([-11, 232.243, 0, 11, -4, 2, 3, 1]), sorted([-11, 232.243, 0, 11, -4, 2, 3, 1]))  x = [random.random() for \_ in range(10)]  self.assertEqual(q\_sort(x), sorted(x))  if \_\_name\_\_ == '\_\_main\_\_':  unittest.main() |

**Приложение В**

Листинг программы тестирования

*Листинг В – Программа тестирования*

|  |
| --- |
| from main import Calculator  import unittest  class TestCalculator(unittest.TestCase):  def setUp(self):  self.calculator = Calculator()  def test\_fib(self):  self.assertEqual(self.calculator.n\_fib(7), 13)  self.assertEqual(self.calculator.n\_fib(4), 3)  self.assertEqual(self.calculator.n\_fib(12), 144)  self.assertEqual(self.calculator.n\_fib(20), 6765)  def test\_fact(self):  self.assertEqual(self.calculator.fact(7), 5040)  self.assertEqual(self.calculator.fact(4), 24)  self.assertEqual(self.calculator.fact(12), 479001600)  self.assertEqual(self.calculator.fact(20), 2432902008176640000)  if \_\_name\_\_ == "\_\_main\_\_":  unittest.main() |