

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт
Информатика
кафедра

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №2

Тестирование программного обеспечения

тема

Преподаватель

Кузнецов Александр Сергеевич

Студент КИ18-16Б 031835768
номер группы, зачетной книжки

До Кармо Флавио
инициалы, фамилия

Красноярск 2021

Цель работы

Вариант 1

Вектор в Евклидовом пространстве

Сравнение

Сложение

Вычитание

Скалярное произведение

Вычисление длины

JSDoc code documentation

Ход работы

Чтобы запустить этот пример на своем компьютере, вам нужно будет следовать инструкциям в следующей репозитории Github:

<https://github.com/drrh12/TDD>

Installing the dependencies

Run “npm i -D jsdoc”

```
Flavi@DESKTOP-2X0BA08 MINGW64 ~/Desktop/UNIVERSITY_2021/Jest-Second-Lab
$ npm i -D jsdoc
npm WARN jest-second-lab@1.0.0 No description
npm WARN jest-second-lab@1.0.0 No repository field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ jsdoc@3.6.7
added 356 packages from 289 contributors and audited 357 packages in 14.237s

28 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Документирование кода с помощью JSDoc

```
1  {
2    "source": {
3      "include": ["src"],
4      "includePatten": ".js$",
5      "excludePatten": "(node_modules|docs)"
6    },
7    "plugins": ["plugins/markdown"],
8    "templates": {
9      "cleverLinks": true,
10     "monospaceLinks": true
11   },
12   "opts": {
13     "recurse": true,
14     "destination": "./docs/"
15   }
16 }
```

jsdoc.json settep and in the package.json a script to run

```
/**
 * Vector
 * @type {Object}
 */
const vectors = {

  /**
   * Comparison
   * @param {Function}
   * @param {Function}
   * @param {Function}
   * This function is used to compare vectors
   */
  comparison: (a, b, c) => {
    const vector = [a, b, c];
    return vector;
  },

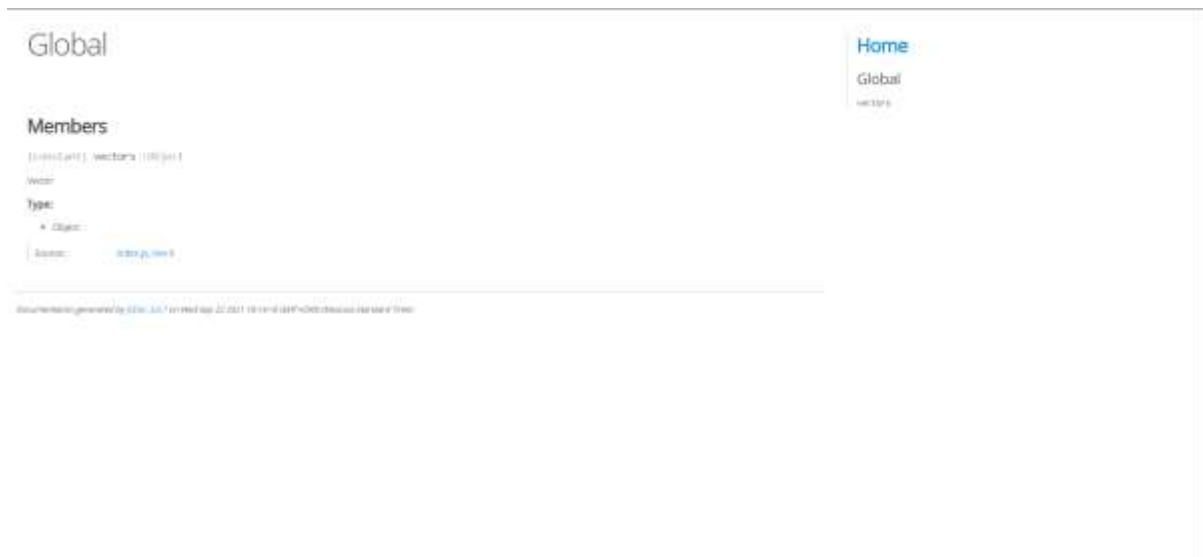
  /**
   * Addition
   * @param {Function}
   * @param {Function}
   * @param {Function}
   * This function is used to add vectors
   */
  addition: (a, b, c) => {
    const first = [a, b, c];
    const second = [1, 2, 3];
    const sum = first.map((a, i) => {
      return a + second[i];
    });
    return sum;
  },

  /**
   * Subtraction
   * @param {Function}
   * @param {Function}
   * @param {Function}
   * This function is used to subtract vectors
   */
  subtraction: (a, b, c) => {
    const first = [a, b, c];
    const second = [1, 2, 3];
    const sub = first.map((a, i) => {
      return a - second[i];
    });
    return sub;
  },

  /**
   * Scalar Product of vector
   * @param {Function}
   * @param {Function}
   * @param {Function}
   * This function is used to get the scalar product of a vector
   */
  scalarProduct: (a, b) => {
    const a = a.map((a, i) => a[i] * b[i]).reduce((a, b) => a + b);
    return a;
  },

  /**
   * Length of vector
   * @param {Function}
   * @param {Function}
   * @param {Function}
   * This function is used to get the length of a vector
   */
  length: (a) => {
    const a = a.map((a, i) => a[i] * a[i]).reduce((a, b) => a + b);
    return Math.sqrt(a);
  }
};
```

Run: npm run doc to generate the html doc.



Source of the file.

Source: index.js

```
1.  /**
2.   * Module
3.   * @type {Object}
4.   */
5.
6. const vectors = {
7.   /**
8.    * Comparison
9.    * @type {Function}
10.    */
11.    /** This function is used to compare vectors */
12.    comparison: () => {
13.      const vector = [4, 5, 8];
14.      return vector;
15.    },
16.    /**
17.     * Addition
18.     * @type {Function}
19.     */
20.    /** This function is used to add vectors */
21.    addition: (a, b, c) => {
22.      const first = [a, b, c];
23.      const second = [1, 2, 3];
24.
25.      const sum = first.map((a, i) => {
26.        return a + second[i];
27.      });
28.      return sum;
29.    }
30.  };
31.  /**
32.   * Main function
33.   * @type {Function}
34.   */
35.  /** This function is used to run the program */
36.  main: () => {
37.    comparison();
38.    addition();
39.  }
40.  /**
41.   * Export the module
42.   * @type {Object}
43.   */
44.  export default vectors;
45.  /**
46.   * Test the module
47.   * @type {Object}
48.   */
49.  test: () => {
50.    comparison();
51.    addition();
52.  }
53.  /**
54.   * Run the program
55.   * @type {Function}
56.   */
57.  run: () => {
58.    main();
59.    test();
60.  }
61.  /**
62.   * Run the program
63.   * @type {Function}
64.   */
65.  run: () => {
66.    main();
67.    test();
68.  }
69.  /**
70.   * Run the program
71.   * @type {Function}
72.   */
73.  run: () => {
74.    main();
75.    test();
76.  }
77.  /**
78.   * Run the program
79.   * @type {Function}
80.   */
81.  run: () => {
82.    main();
83.    test();
84.  }
85.  /**
86.   * Run the program
87.   * @type {Function}
88.   */
89.  run: () => {
90.    main();
91.    test();
92.  }
93.  /**
94.   * Run the program
95.   * @type {Function}
96.   */
97.  run: () => {
98.    main();
99.    test();
100.  }
101.  /**
102.   * Run the program
103.   * @type {Function}
104.   */
105.  run: () => {
106.    main();
107.    test();
108.  }
109.  /**
110.   * Run the program
111.   * @type {Function}
112.   */
113.  run: () => {
114.    main();
115.    test();
116.  }
117.  /**
118.   * Run the program
119.   * @type {Function}
120.   */
121.  run: () => {
122.    main();
123.    test();
124.  }
125.  /**
126.   * Run the program
127.   * @type {Function}
128.   */
129.  run: () => {
130.    main();
131.    test();
132.  }
133.  /**
134.   * Run the program
135.   * @type {Function}
136.   */
137.  run: () => {
138.    main();
139.    test();
140.  }
141.  /**
142.   * Run the program
143.   * @type {Function}
144.   */
145.  run: () => {
146.    main();
147.    test();
148.  }
149.  /**
150.   * Run the program
151.   * @type {Function}
152.   */
153.  run: () => {
154.    main();
155.    test();
156.  }
157.  /**
158.   * Run the program
159.   * @type {Function}
160.   */
161.  run: () => {
162.    main();
163.    test();
164.  }
165.  /**
166.   * Run the program
167.   * @type {Function}
168.   */
169.  run: () => {
170.    main();
171.    test();
172.  }
173.  /**
174.   * Run the program
175.   * @type {Function}
176.   */
177.  run: () => {
178.    main();
179.    test();
180.  }
181.  /**
182.   * Run the program
183.   * @type {Function}
184.   */
185.  run: () => {
186.    main();
187.    test();
188.  }
189.  /**
190.   * Run the program
191.   * @type {Function}
192.   */
193.  run: () => {
194.    main();
195.    test();
196.  }
197.  /**
198.   * Run the program
199.   * @type {Function}
200.   */
201.  run: () => {
202.    main();
203.    test();
204.  }
205.  /**
206.   * Run the program
207.   * @type {Function}
208.   */
209.  run: () => {
210.    main();
211.    test();
212.  }
213.  /**
214.   * Run the program
215.   * @type {Function}
216.   */
217.  run: () => {
218.    main();
219.    test();
220.  }
221.  /**
222.   * Run the program
223.   * @type {Function}
224.   */
225.  run: () => {
226.    main();
227.    test();
228.  }
229.  /**
230.   * Run the program
231.   * @type {Function}
232.   */
233.  run: () => {
234.    main();
235.    test();
236.  }
237.  /**
238.   * Run the program
239.   * @type {Function}
240.   */
241.  run: () => {
242.    main();
243.    test();
244.  }
245.  /**
246.   * Run the program
247.   * @type {Function}
248.   */
249.  run: () => {
250.    main();
251.    test();
252.  }
253.  /**
254.   * Run the program
255.   * @type {Function}
256.   */
257.  run: () => {
258.    main();
259.    test();
260.  }
261.  /**
262.   * Run the program
263.   * @type {Function}
264.   */
265.  run: () => {
266.    main();
267.    test();
268.  }
269.  /**
270.   * Run the program
271.   * @type {Function}
272.   */
273.  run: () => {
274.    main();
275.    test();
276.  }
277.  /**
278.   * Run the program
279.   * @type {Function}
280.   */
281.  run: () => {
282.    main();
283.    test();
284.  }
285.  /**
286.   * Run the program
287.   * @type {Function}
288.   */
289.  run: () => {
290.    main();
291.    test();
292.  }
293.  /**
294.   * Run the program
295.   * @type {Function}
296.   */
297.  run: () => {
298.    main();
299.    test();
300.  }
301.  /**
302.   * Run the program
303.   * @type {Function}
304.   */
305.  run: () => {
306.    main();
307.    test();
308.  }
309.  /**
310.   * Run the program
311.   * @type {Function}
312.   */
313.  run: () => {
314.    main();
315.    test();
316.  }
317.  /**
318.   * Run the program
319.   * @type {Function}
320.   */
321.  run: () => {
322.    main();
323.    test();
324.  }
325.  /**
326.   * Run the program
327.   * @type {Function}
328.   */
329.  run: () => {
330.    main();
331.    test();
332.  }
333.  /**
334.   * Run the program
335.   * @type {Function}
336.   */
337.  run: () => {
338.    main();
339.    test();
340.  }
341.  /**
342.   * Run the program
343.   * @type {Function}
344.   */
345.  run: () => {
346.    main();
347.    test();
348.  }
349.  /**
350.   * Run the program
351.   * @type {Function}
352.   */
353.  run: () => {
354.    main();
355.    test();
356.  }
357.  /**
358.   * Run the program
359.   * @type {Function}
360.   */
361.  run: () => {
362.    main();
363.    test();
364.  }
365.  /**
366.   * Run the program
367.   * @type {Function}
368.   */
369.  run: () => {
370.    main();
371.    test();
372.  }
373.  /**
374.   * Run the program
375.   * @type {Function}
376.   */
377.  run: () => {
378.    main();
379.    test();
380.  }
381.  /**
382.   * Run the program
383.   * @type {Function}
384.   */
385.  run: () => {
386.    main();
387.    test();
388.  }
389.  /**
390.   * Run the program
391.   * @type {Function}
392.   */
393.  run: () => {
394.    main();
395.    test();
396.  }
397.  /**
398.   * Run the program
399.   * @type {Function}
400.   */
401.  run: () => {
402.    main();
403.    test();
404.  }
405.  /**
406.   * Run the program
407.   * @type {Function}
408.   */
409.  run: () => {
410.    main();
411.    test();
412.  }
413.  /**
414.   * Run the program
415.   * @type {Function}
416.   */
417.  run: () => {
418.    main();
419.    test();
420.  }
421.  /**
422.   * Run the program
423.   * @type {Function}
424.   */
425.  run: () => {
426.    main();
427.    test();
428.  }
429.  /**
430.   * Run the program
431.   * @type {Function}
432.   */
433.  run: () => {
434.    main();
435.    test();
436.  }
437.  /**
438.   * Run the program
439.   * @type {Function}
440.   */
441.  run: () => {
442.    main();
443.    test();
444.  }
445.  /**
446.   * Run the program
447.   * @type {Function}
448.   */
449.  run: () => {
450.    main();
451.    test();
452.  }
453.  /**
454.   * Run the program
455.   * @type {Function}
456.   */
457.  run: () => {
458.    main();
459.    test();
460.  }
461.  /**
462.   * Run the program
463.   * @type {Function}
464.   */
465.  run: () => {
466.    main();
467.    test();
468.  }
469.  /**
470.   * Run the program
471.   * @type {Function}
472.   */
473.  run: () => {
474.    main();
475.    test();
476.  }
477.  /**
478.   * Run the program
479.   * @type {Function}
480.   */
481.  run: () => {
482.    main();
483.    test();
484.  }
485.  /**
486.   * Run the program
487.   * @type {Function}
488.   */
489.  run: () => {
490.    main();
491.    test();
492.  }
493.  /**
494.   * Run the program
495.   * @type {Function}
496.   */
497.  run: () => {
498.    main();
499.    test();
500.  }
501.  /**
502.   * Run the program
503.   * @type {Function}
504.   */
505.  run: () => {
506.    main();
507.    test();
508.  }
509.  /**
510.   * Run the program
511.   * @type {Function}
512.   */
513.  run: () => {
514.    main();
515.    test();
516.  }
517.  /**
518.   * Run the program
519.   * @type {Function}
520.   */
521.  run: () => {
522.    main();
523.    test();
524.  }
525.  /**
526.   * Run the program
527.   * @type {Function}
528.   */
529.  run: () => {
530.    main();
531.    test();
532.  }
533.  /**
534.   * Run the program
535.   * @type {Function}
536.   */
537.  run: () => {
538.    main();
539.    test();
540.  }
541.  /**
542.   * Run the program
543.   * @type {Function}
544.   */
545.  run: () => {
546.    main();
547.    test();
548.  }
549.  /**
550.   * Run the program
551.   * @type {Function}
552.   */
553.  run: () => {
554.    main();
555.    test();
556.  }
557.  /**
558.   * Run the program
559.   * @type {Function}
560.   */
561.  run: () => {
562.    main();
563.    test();
564.  }
565.  /**
566.   * Run the program
567.   * @type {Function}
568.   */
569.  run: () => {
570.    main();
571.    test();
572.  }
573.  /**
574.   * Run the program
575.   * @type {Function}
576.   */
577.  run: () => {
578.    main();
579.    test();
580.  }
581.  /**
582.   * Run the program
583.   * @type {Function}
584.   */
585.  run: () => {
586.    main();
587.    test();
588.  }
589.  /**
590.   * Run the program
591.   * @type {Function}
592.   */
593.  run: () => {
594.    main();
595.    test();
596.  }
597.  /**
598.   * Run the program
599.   * @type {Function}
600.   */
601.  run: () => {
602.    main();
603.    test();
604.  }
605.  /**
606.   * Run the program
607.   * @type {Function}
608.   */
609.  run: () => {
610.    main();
611.    test();
612.  }
613.  /**
614.   * Run the program
615.   * @type {Function}
616.   */
617.  run: () => {
618.    main();
619.    test();
620.  }
621.  /**
622.   * Run the program
623.   * @type {Function}
624.   */
625.  run: () => {
626.    main();
627.    test();
628.  }
629.  /**
630.   * Run the program
631.   * @type {Function}
632.   */
633.  run: () => {
634.    main();
635.    test();
636.  }
637.  /**
638.   * Run the program
639.   * @type {Function}
640.   */
641.  run: () => {
642.    main();
643.    test();
644.  }
645.  /**
646.   * Run the program
647.   * @type {Function}
648.   */
649.  run: () => {
650.    main();
651.    test();
652.  }
653.  /**
654.   * Run the program
655.   * @type {Function}
656.   */
657.  run: () => {
658.    main();
659.    test();
660.  }
661.  /**
662.   * Run the program
663.   * @type {Function}
664.   */
665.  run: () => {
666.    main();
667.    test();
668.  }
669.  /**
670.   * Run the program
671.   * @type {Function}
672.   */
673.  run: () => {
674.    main();
675.    test();
676.  }
677.  /**
678.   * Run the program
679.   * @type {Function}
680.   */
681.  run: () => {
682.    main();
683.    test();
684.  }
685.  /**
686.   * Run the program
687.   * @type {Function}
688.   */
689.  run: () => {
690.    main();
691.    test();
692.  }
693.  /**
694.   * Run the program
695.   * @type {Function}
696.   */
697.  run: () => {
698.    main();
699.    test();
700.  }
701.  /**
702.   * Run the program
703.   * @type {Function}
704.   */
705.  run: () => {
706.    main();
707.    test();
708.  }
709.  /**
710.   * Run the program
711.   * @type {Function}
712.   */
713.  run: () => {
714.    main();
715.    test();
716.  }
717.  /**
718.   * Run the program
719.   * @type {Function}
720.   */
721.  run: () => {
722.    main();
723.    test();
724.  }
725.  /**
726.   * Run the program
727.   * @type {Function}
728.   */
729.  run: () => {
730.    main();
731.    test();
732.  }
733.  /**
734.   * Run the program
735.   * @type {Function}
736.   */
737.  run: () => {
738.    main();
739.    test();
740.  }
741.  /**
742.   * Run the program
743.   * @type {Function}
744.   */
745.  run: () => {
746.    main();
747.    test();
748.  }
749.  /**
750.   * Run the program
751.   * @type {Function}
752.   */
753.  run: () => {
754.    main();
755.    test();
756.  }
757.  /**
758.   * Run the program
759.   * @type {Function}
760.   */
761.  run: () => {
762.    main();
763.    test();
764.  }
765.  /**
766.   * Run the program
767.   * @type {Function}
768.   */
769.  run: () => {
770.    main();
771.    test();
772.  }
773.  /**
774.   * Run the program
775.   * @type {Function}
776.   */
777.  run: () => {
778.    main();
779.    test();
780.  }
781.  /**
782.   * Run the program
783.   * @type {Function}
784.   */
785.  run: () => {
786.    main();
787.    test();
788.  }
789.  /**
790.   * Run the program
791.   * @type {Function}
792.   */
793.  run: () => {
794.    main();
795.    test();
796.  }
797.  /**
798.   * Run the program
799.   * @type {Function}
800.   */
801.  run: () => {
802.    main();
803.    test();
804.  }
805.  /**
806.   * Run the program
807.   * @type {Function}
808.   */
809.  run: () => {
810.    main();
811.    test();
812.  }
813.  /**
814.   * Run the program
815.   * @type {Function}
816.   */
817.  run: () => {
818.    main();
819.    test();
820.  }
821.  /**
822.   * Run the program
823.   * @type {Function}
824.   */
825.  run: () => {
826.    main();
827.    test();
828.  }
829.  /**
830.   * Run the program
831.   * @type {Function}
832.   */
833.  run: () => {
834.    main();
835.    test();
836.  }
837.  /**
838.   * Run the program
839.   * @type {Function}
840.   */
841.  run: () => {
842.    main();
843.    test();
844.  }
845.  /**
846.   * Run the program
847.   * @type {Function}
848.   */
849.  run: () => {
850.    main();
851.    test();
852.  }
853.  /**
854.   * Run the program
855.   * @type {Function}
856.   */
857.  run: () => {
858.    main();
859.    test();
860.  }
861.  /**
862.   * Run the program
863.   * @type {Function}
864.   */
865.  run: () => {
866.    main();
867.    test();
868.  }
869.  /**
870.   * Run the program
871.   * @type {Function}
872.   */
873.  run: () => {
874.    main();
875.    test();
876.  }
877.  /**
878.   * Run the program
879.   * @type {Function}
880.   */
881.  run: () => {
882.    main();
883.    test();
884.  }
885.  /**
886.   * Run the program
887.   * @type {Function}
888.   */
889.  run: () => {
890.    main();
891.    test();
892.  }
893.  /**
894.   * Run the program
895.   * @type {Function}
896.   */
897.  run: () => {
898.    main();
899.    test();
900.  }
901.  /**
902.   * Run the program
903.   * @type {Function}
904.   */
905.  run: () => {
906.    main();
907.    test();
908.  }
909.  /**
910.   * Run the program
911.   * @type {Function}
912.   */
913.  run: () => {
914.    main();
915.    test();
916.  }
917.  /**
918.   * Run the program
919.   * @type {Function}
920.   */
921.  run: () => {
922.    main();
923.    test();
924.  }
925.  /**
926.   * Run the program
927.   * @type {Function}
928.   */
929.  run: () => {
930.    main();
931.    test();
932.  }
933.  /**
934.   * Run the program
935.   * @type {Function}
936.   */
937.  run: () => {
938.    main();
939.    test();
940.  }
941.  /**
942.   * Run the program
943.   * @type {Function}
944.   */
945.  run: () => {
946.    main();
947.    test();
948.  }
949.  /**
950.   * Run the program
951.   * @type {Function}
952.   */
953.  run: () => {
954.    main();
955.    test();
956.  }
957.  /**
958.   * Run the program
959.   * @type {Function}
960.   */
961.  run: () => {
962.    main();
963.    test();
964.  }
965.  /**
966.   * Run the program
967.   * @type {Function}
968.   */
969.  run: () => {
970.    main();
971.    test();
972.  }
973.  /**
974.   * Run the program
975.   * @type {Function}
976.   */
977.  run: () => {
978.    main();
979.    test();
980.  }
981.  /**
982.   * Run the program
983.   * @type {Function}
984.   */
985.  run: () => {
986.    main();
987.    test();
988.  }
989.  /**
990.   * Run the program
991.   * @type {Function}
992.   */
993.  run: () => {
994.    main();
995.    test();
996.  }
997.  /**
998.   * Run the program
999.   * @type {Function}
1000.   */
1001.  run: () => {
1002.    main();
1003.    test();
1004.  }
1005.  /**
1006.   * Run the program
1007.   * @type {Function}
1008.   */
1009.  run: () => {
1010.    main();
1011.    test();
1012.  }
1013.  /**
1014.   * Run the program
1015.   * @type {Function}
1016.   */
1017.  run: () => {
1018.    main();
1019.    test();
1020.  }
1021.  /**
1022.   * Run the program
1023.   * @type {Function}
1024.   */
1025.  run: () => {
1026.    main();
1027.    test();
1028.  }
1029.  /**
1030.   * Run the program
1031.   * @type {Function}
1032.   */
1033.  run: () => {
1034.    main();
1035.    test();
1036.  }
1037.  /**
1038.   * Run the program
1039.   * @type {Function}
1040.   */
1041.  run: () => {
1042.    main();
1043.    test();
1044.  }
1045.  /**
1046.   * Run the program
1047.   * @type {Function}
1048.   */
1049.  run: () => {
1050.    main();
1051.    test();
1052.  }
1053.  /**
1054.   * Run the program
1055.   * @type {Function}
1056.   */
1057.  run: () => {
1058.    main();
1059.    test();
1060.  }
1061.  /**
1062.   * Run the program
1063.   * @type {Function}
1064.   */
1065.  run: () => {
1066.    main();
1067.    test();
1068.  }
1069.  /**
1070.   * Run the program
1071.   * @type {Function}
1072.   */
1073.  run: () => {
1074.    main();
1075.    test();
1076.  }
1077.  /**
1078.   * Run the program
1079.   * @type {Function}
1080.   */
1081.  run: () => {
1082.    main();
1083.    test();
1084.  }
1085.  /**
1086.   * Run the program
1087.   * @type {Function}
1088.   */
1089.  run: () => {
1090.    main();
1091.    test();
1092.  }
1093.  /**
1094.   * Run the program
1095.   * @type {Function}
1096.   */
1097.  run: () => {
1098.    main();
1099.    test();
1100.  }
1101.  /**
1102.   * Run the program
1103.   * @type {Function}
1104.   */
1105.  run: () => {
1106.    main();
1107.    test();
1108.  }
1109.  /**
1110.   * Run the program
1111.   * @type {Function}
1112.   */
1113.  run: () => {
1114.    main();
1115.    test();
1116.  }
1117.  /**
1118.   * Run the program
1119.   * @type {Function}
1120.   */
1121.  run: () => {
1122.    main();
1123.    test();
1124.  }
1125.  /**
1126.   * Run the program
1127.   * @type {Function}
1128.   */
1129.  run: () => {
1130.    main();
1131.    test();
1132.  }
1133.  /**
1134.   * Run the program
1135.   * @type {Function}
1136.   */
1137.  run: () => {
1138.    main();
1139.    test();
1140.  }
1141.  /**
1142.   * Run the program
1143.   * @type {Function}
1144.   */
1145.  run: () => {
1146.    main();
1147.    test();
1148.  }
1149.  /**
1150.   * Run the program
1151.   * @type {Function}
1152.   */
1153.  run: () => {
1154.    main();
1155.    test();
1156.  }
1157.  /**
1158.   * Run the program
1159.   * @type {Function}
1160.   */
1161.  run: () => {
1162.    main();
1163.    test();
1164.  }
1165.  /**
1166.   * Run the program
1167.   * @type {Function}
1168.   */
1169.  run: () => {
1170.    main();
1171.    test();
1172.  }
1173.  /**
1174.   * Run the program
1175.   * @type {Function}
1176.   */
1177.  run: () => {
1178.    main();
1179.    test();
1180.  }
1181.  /**
1182.   * Run the program
1183.   * @type {Function}
1184.   */
1185.  run: () => {
1186.    main();
1187.    test();
1188.  }
1189.  /**
1190.   * Run the program
1191.   * @type {Function}
1192.   */
1193.  run: () => {
1194.    main();
1195.    test();
1196.  }
1197.  /**
1198.   * Run the program
1199.   * @type {Function}
1200.   */
1201.  run: () => {
1202.    main();
1203.    test();
1204.  }
1205.  /**
1206.   * Run the program
1207.   * @type {Function}
1208.   */
1209.  run: () => {
1210.    main();
1211.    test();
1212.  }
1213.  /**
1214.   * Run the program
1215.   * @type {Function}
1216.   */
1217.  run: () => {
1218.    main();
1219.    test();
1220.  }
1221.  /**
1222.   * Run the program
1223.   * @type {Function}
1224.   */
1225.  run: () => {
1226.    main();
1227.    test();
1228.  }
1229.  /**
1230.   * Run the program
1231.   * @type {Function}
1232.   */
1233.  run: () => {
1234.    main();
1235.    test();
1236.  }
1237.  /**
1238.   * Run the program
1239.   * @type {Function}
1240.   */
1241.  run: () => {
1242.    main();
1243.    test();
1244.  }
1245.  /**
1246.   * Run the program
1247.   * @type {Function}
1248.   */
1249.  run: () => {
1250.    main();
1251.    test();
1252.  }
1253.  /**
1254.   * Run the program
1255.   * @type {Function}
1256.   */
1257.  run: () => {
1258.    main();
1259.    test();
1260.  }
1261.  /**
1262.   * Run the program
1263.   * @type {Function}
1264.   */
1265.  run: () => {
1266.    main();
1267.    test();
1268.  }
1269.  /**
1270.   * Run the program
1271.   * @type {Function}
1272.   */
1273.  run: () => {
1274.    main();
1275.    test();
1276.  }
1277.  /**
1278.   * Run the program
1279.   * @type {Function}
1280.   */
1281.  run: () => {
1282.    main();
1283.    test();
1284.  }
1285.  /**
1286.   * Run the program
1287.   * @type {Function}
1288.   */
1289.  run: () => {
1290.    main();
1291.    test();
1292.  }
1293.  /**
1294.   * Run the program
1295.   * @type {Function}
1296.   */
1297.  run: () => {
1298.    main();
1299.    test();
1300.  }
1301.  /**
1302.   * Run the program
1303.   * @type {Function}
1304.   */
1305.  run: () => {
1306.    main();
1307.    test
```

```

C:\Users\user> cd C:\Users\user\Documents\UNIVERSITY 2021\test-second-lab
& npm test

> test-second-lab@0.0.0 test C:\Users\user\Documents\UNIVERSITY 2021\test-second-lab
> jest --coverage

  console.log
    Comparing the equivalence of two vectors (3 ms)
    +-----+
    |File|      % Tests | % Branch | % Pass | % Fail | Expected Code Pk |
    +-----+
    |All Files|      100 |      100 |    100 |    0% |          0/0/0/0 |
    |Function.test.js| 100 |      100 |    100 |    0% |          0/0/0/0 |
    +-----+

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        0.416 s, estimated 1 s
Ran all test suites.

```

2. Сложение

```

const vectors = {
  comparison: () => {
    const vector = [4, 5, 8];
    return vector;
  },
  addition: (a, b, c) => {
    const first = [a, b, c];
    const second = [1, 3, 2];

    const sum = first.map((n, i) => {
      return n + second[i];
    });
    return sum;
  },
};

test("Addition of vectors", () => {
  expect(vectors.addition(2, 1, 5)).toEqual([3, 4, 7]);
});

```

Test:

```

C:\Users\user> cd C:\Users\user\Documents\UNIVERSITY 2021\test-second-lab
& npm test

> test-second-lab@0.0.0 test C:\Users\user\Documents\UNIVERSITY 2021\test-second-lab
> jest --coverage

  console.log
    Comparing the equivalence of two vectors (3 ms)
    +-----+
    |File|      % Tests | % Branch | % Pass | % Fail | Expected Code Pk |
    +-----+
    |All Files|      100 |      100 |    100 |    0% |          0/0/0/0 |
    |Function.test.js| 100 |      100 |    100 |    0% |          0/0/0/0 |
    +-----+

Test Suites: 1 passed, 1 total
Tests:       3 passed, 3 total
Snapshots:   0 total
Time:        0.416 s, estimated 1 s
Ran all test suites.

```

3. Вычитание

Functions

```
const vectors = {  
  comparison: () => {  
    const vector = [4, 5, 8];  
    return vector;  
  },  
  addition: (a, b, c) => {  
    const first = [a, b, c];  
    const second = [1, 3, 2];  
  
    const sum = first.map((n, i) => {  
      return n + second[i];  
    });  
    return sum;  
  },  
  subtraction: (a, b, c) => {  
    const first = [a, b, c];  
    const second = [1, 3, 2];  
  
    const sum = first.map((n, i) => {  
      return n - second[i];  
    });  
    return sum;  
  },  
};
```

Test:

```
npm test  
  
test second-4.4.4 test C:\Users\Ivan\Documents\second-4.4.4  
  
- Functions.test.js  
  - Creating the representation of two vectors (2 to)  
    - Addition of vectors  
    - Subtraction of vectors  
  
File | S Status | S Branch | S Pass | S Fail | S Lines | Executed Line No.  
-----  
All Files | 66.67 | 100 | 66.67 | 0.00 | 66.67 |  
Functions.js | 66.67 | 100 | 66.67 | 0.00 | 66.67 | 1-4, 23-30  
  
Test Suites: 1 passed, 1 total  
Tests: 4 passed, 4 total  
Snapshots: 0 total  
Time: 0.105 s, estimated 1 s  
Run all test suites.
```

4. Скалярное произведение

```
const vectors = {  
  comparison: () => {  
    const vector = [4, 5, 8];  
    return vector;  
  },  
  addition: (a, b, c) => {  
    const first = [a, b, c];  
    const second = [1, 3, 2];  
  
    const sum = first.map((n, i) => {  
      return n + second[i];  
    });  
    return sum;  
  },  
  subtraction: (a, b, c) => {  
    const first = [a, b, c];  
    const second = [1, 3, 2];  
  
    const sum = first.map((n, i) => {  
      return n - second[i];  
    });  
    return sum;  
  },  
  scalarProduct: (a, b) => {  
    const c = a.map((x, i) => a[i] * b[i]).reduce((m, n) => m + n);  
    return c;  
  },  
};
```

Testing:

```

$ npm test

> jest-second-lab@1.0.0 test C:\Users\Flavi\Desktop\UNIVERSITY 2021\Jest-Second-Lab
> jest --coverage

PASS ./functions.test.js
  ✓ Comparing the equivalence of two vectors (2 ms)
  ✓ Addition of vectors (1 ms)
  ✓ Subtraction of vectors (1 ms)
  ✓ Scalar product of vectors
  ✓ Length of vectors

at Object.runCommand [Function] (node:12:14)

-----|-----|-----|-----|-----|
File      | % Stats | % Branch | % Funcs | % Lines | Uncovered Line #s |
-----|-----|-----|-----|-----|
All files | 100     | 100      | 100     | 100     |                    |
functions.js | 100    | 100     | 100     | 100     |                    |
-----|-----|-----|-----|-----|
Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        0.107 s, estimated 1 s
Ran all test suites.

```

5. Вычисление длины

```

const vectors = {
  comparison: () => {
    const vector = [4, 5, 8];
    return vector;
  },
  addition: (a, b, c) => {
    const first = [a, b, c];
    const second = [1, 3, 2];

    const sum = first.map((n, i) => {
      return n + second[i];
    });
    return sum;
  },
  subtraction: (a, b, c) => {
    const first = [a, b, c];
    const second = [1, 3, 2];

    const sum = first.map((n, i) => {
      return n - second[i];
    });
    return sum;
  },
  scalarProduct: (a, b) => {
    const c = a.map((x, i) => a[i] * b[i]).reduce((m, n) => m + n);
    return c;
  },
  length: (a) => {
    const l = a.length;
    return l;
  },
};

```

Test

```

flavi@DESKTOP-2K0BAD0 MINGW64 ~/Desktop/UNIVERSITY 2021/Jest-Second-Lab
$ npm test

> jest-second-lab@1.0.0 test C:\Users\Flavi\Desktop\UNIVERSITY 2021\Jest-Second-Lab
> jest --coverage

PASS ./functions.test.js
  ✓ Comparing the equivalence of two vectors (2 ms)
  ✓ Addition of vectors (1 ms)
  ✓ Subtraction of vectors (1 ms)
  ✓ Scalar product of vectors (3 ms)
  ✓ Length of vectors

-----|-----|-----|-----|-----|
File      | % Stats | % Branch | % Funcs | % Lines | Uncovered Line #s |
-----|-----|-----|-----|-----|
All files | 100     | 100      | 100     | 100     |                    |
functions.js | 100    | 100     | 100     | 100     |                    |
-----|-----|-----|-----|-----|
Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        0.445 s, estimated 1 s
Ran all test suites.

```

Заключение:

В этом упражнении было изучено и изучено, что такое модульный тест, приспособления для тестирования и параметризованное тестирование, а также способы его реализации с использованием JavaScript и Jest Framework.

Документация и источники, используемые для понимания того, что такое модульное тестирование, приспособления для тестирования и параметризованные тесты.

<https://jestjs.io/docs/getting-started>

<https://medium.com/@suman.maity112/parameterized-tests-8929080901ee>

<https://medium.com/@suman.maity112/parameterized-tests-with-jest-2932f42b1509>

https://www.youtube.com/watch?v=7r4xVDI2vho&t=242s&ab_channel=TraversyMedia

<https://jsdoc.app/>

<https://jsdoc.app/about-getting-started.html>