

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа «Программная инженерия»

СОГЛАСОВАНО
Доцент департамента
программной инженерии
факультета компьютерных наук
канд. техн. наук

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук

И.Ю. Самоненко
«__» _____ 2022 г.

В.В.Шилов
«__» _____ 2022 г.

ПРОГРАММА ВИЗУАЛИЗАЦИИ АЛГОРИТМОВ И СТРУКТУР ДАННЫХ

Программа и методика испытаний

**ЛИСТ УТВЕРЖДЕНИЯ
RU.17701729.10.03-01 51 01-1-ЛУ**

Исполнитель
студент группы БПИ203
_____/ П.Н.Ломакин/
«__» _____ 2022 г.

УТВЕРЖДЕН
RU.17701729.10.03-01 51 01-1-ЛУ

ПРОГРАММА ВИЗУАЛИЗАЦИИ АЛГОРИТМОВ И СТРУКТУР ДАННЫХ

Программа и методика испытаний

RU.17701729.10.03-01 51 01-1-ЛУ

Листов 11

1. ОБЪЕКТ ИСПЫТАНИЙ

1.1 Наименование

Наименование программы – “Программа визуализации алгоритмов и структур данных”.

Наименование программы на английском языке – “ Program for visualization of algorithms and data structures” .

1.2 Область применения

Данное приложение полезно как для студентов, которые проходят курс по алгоритмам, так и преподавателям, которые эти курсы ведут.

1.3 Обозначение испытуемой программы

Краткое наименование программы – “VADS”.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. ЦЕЛЬ ИСПЫТАНИЙ

Целью проведения испытаний является проверка соответствия характеристик разработанной программы функциональным требованиям и отдельным требованиям к надежности, изложенных в документе Техническое задание к данной программе.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. ТРЕБОВАНИЯ К ПРОГРАММЕ

3.1 Требования к функциональным характеристикам

3.1.1 Требования к составу выполняемых функций

Программа должна выполнять следующие функции;

- 1) Выбор из списка алгоритмов / структур данных интересующий.
- 2) Возможность прочитать краткое описание, которое поможет глубже разобраться с алгоритмом / структурой данных.
- 3) Показать процесс работы алгоритма / построения структур данных.
- 4) Следить за псевдокодом во время визуализации.
- 5) Давать пользователю возможность изменять входные данные для каждого алгоритма / структур данных.
- 6) Изменение скорости подачи материала.
- 7) Есть специальных раздел, в котором есть возможность посмотреть за поведением алгоритма / структуры данных при различных входных параметрах (рисуется график).

В данном приложении в качестве примера будет реализована структура данных “Фильтр Блума”.

3.1.2 Требования к организации входных данных

Обработка входных данных, введенных пользователем в соответствующие поля для ввода.

3.1.3 Требования к организации выходных данных

В качестве выходных данных используется визуализация того или иного алгоритма / структуры данных.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.2 Требования к надежности

Приложение должно стабильно работать при применении по назначению, при соблюдении условий эксплуатации, требований к составу и параметрам технических средств и отсутствии редактирования или повреждения файлов.

3.2 Требования к интерфейсу

- 1) Программа является web-приложением.
- 2) Интерфейс интуитивно понятен пользователю.
- 3) Приложение должно правильно считывать входные параметры, введенные пользователем.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

4.1 Состав программной документации

Состав программной документации должен включать в себя следующие компоненты:

- 1) Техническое задание (ГОСТ 19.201-78);
- 2) Программа и методика испытаний (ГОСТ 19.301-78);
- 3) Пояснительная записка (ГОСТ 19.404-79);
- 4) Руководство оператора (ГОСТ 19.505-79);
- 5) Текст программы (ГОСТ 19.401-78).

4.2 Специальные требования к программной документации

Все документы к программе выполняются в соответствии с ГОСТ 19.106-78 и ГОСТ к этому виду документа (см. п. 4.1.).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ

5.1 Программные средства, используемые во время испытаний

Список браузеров, которые требуется использовать для корректной работы программы: Safari, Google Chrome.

5.2 Порядок проведения испытаний

Испытания должны проводиться в следующем порядке:

- 1) проверка требований к программной документации;
- 2) проверка требований к интерфейсу;
- 3) проверка требований к функциональным характеристикам;
- 4) проверка требований к надежности.

5.3 Условия проведения испытаний

5.3.1 Климатические условия

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к персональным компьютерам и внешним USB-накопителям в части условий их эксплуатации.

5.3.2 Требования к численности и квалификации персонала

Для испытаний программы требуется по крайней мере один пользователь. Пользователь не должен обладать особыми навыками.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

6. МЕТОДЫ ИСПЫТАНИЙ

6.1 Испытание выполнения требований к программной документации

Соответствие программной документации требованиям проверяется путем просмотра программной документации вручную.
Путем просмотра выявлено, что программная документация удовлетворяет требованиям.

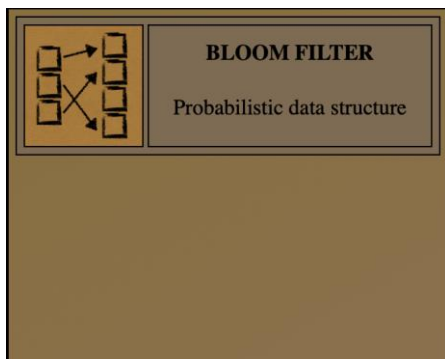
6.2 Испытание выполнения требований к интерфейсу

Программа работает на платформе Windows, macOS и Linux, имеет удобный и понятный интерфейс.

6.3 Испытание выполнения требований к функциональным характеристикам

6.3.1 Испытание выполнения требований к выбору доступного алгоритма / структуры данных.

В главном меню пользователю доступен выбор между алгоритмами и структура данных:



В данном случае доступна только структура данных Фильтр Блума.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

6.3.2 Испытание выполнения требований к возможности изменять входные данные

После того, как пользователь выбран алгоритм / структуру данных (в данном приложении доступен только Фильтр Блума), то он имеет возможность ввести входные данные:

Filter size: restrictions: $1 \leq \text{value} \leq 13$

Number of hash: restrictions: $1 \leq \text{value} \leq 13$

BUILD FILTER

При различных входных данных приложение работает корректно.

6.4.2 Испытание выполнения требований к возможности изменять настройки приложения

У пользователя есть возможность изменять некоторые настройки приложения (в данном приложении доступен только Фильтр Блума). Данные настройки работают корректно.

Speed visualization: faster slower

When hovering over a hash function, you can see how the final value is calculated!

6.4.3 Испытание выполнения требований к возможности визуализировать алгоритм / структуру данных

Визуализация алгоритма / структуры данных работает корректно при различных входных данных:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

VADS

BLOOM FILTER

Probabilistic data structure

Description

The Bloom filter is a bitmap of m bits. At the very beginning, the user defines k independent hash functions, each of which is mapped to a filter, that is, to a bit array. If the function mapped to the i -th cell, the i -th bit becomes equal to **one**.
Optimal number of hash functions: $(m/n)\ln 2 \approx 0.6931(m/n)$.

Use cases

Bloom filter allows you to determine that the element is **not** included in the set. For example, if a database is accessed, then the filter can say with **100%** probability that the element is **not** there.

Output cases

There are **three** possible outputs: the element is **definitely not** in the filter, the element **may be** present, or a **false positive** (occurs when the filter says that the element may be present, but it was not added).

Probabilities

False positive probability: $(1 - e^{-kn/m})^k$
False alarm probability: $2^{-k} \approx 0.6185^{m/n}$,
where k - number of functions, m - filter size,
 n - number of added elements.

Filter size: restrictions: 1 ≤ value ≤ 13

Number of hash: restrictions: 1 ≤ value ≤ 13

BUILD FILTER

1: 2

Added elements

Add element

 Check availability

k₁
k₂
k₃

1
0
1
0
1
0
0

Clear filter

Change function parameters

Accept changes

Change randomly

Speed visualization:
faster
slower

When hovering over a hash function, you can see how the final value is calculated!

Input data analysis

Filter Bloom size:

Number of hash functions:

Number of elements to add:

Number of elements to check:

Range for elements:

generate

First, a filter is created, after which randomly generated numbers are added to it, the number of which is equal to the value in the third field.

After that, new numbers are randomly generated (their number is equal to the value in the fourth field), which are checked for belonging to the filter.

Category	Value
false positive	15
negative	18
positive	16

```

int element = readQ();
for (int index = 0; index < numberOfFunctions; ++index) {
    int indexInFilter = listOfFunctions[index](element);
    --> filter indexInFilter == 1
}
          
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ 1

Список использованной литературы

- 1) ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 2) ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 3) ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 4) ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 5) ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 7) ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 8) ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 9) ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.V
- 10) VisuAlgo. [Электронный ресурс]//URL: <https://visualgo.net/ru> (Дата обращения: 03.02.2022, режим доступа: свободный).
- 11) Vamonos. [Электронный ресурс]//URL: <https://rosulek.github.io/vamonos/> (Дата обращения: 03.02.2022, режим доступа: свободный).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

[illegible]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата