# Практическое задание 1

Тема «Лексический анализ программ»

Задание. Необходимо разработать программу для вычисления значений функции на интервале от *xn* до *xk* с шагом *h* (см. табл. 2) и оценить качество программы с использованием метрик Джилба на основе лексического анализа. Пусть *a*, *b, c* – действительные числа.

*Номер варианта выбирается из таблицы 1 по первой букве фамилии студента.*

Таблица 1

Таблица выбора номера варианта

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Первая буква фамилии студента | А  К  Ф | Б  Л  Х | В  М  Ц | Г  Н  Ч | Д  О  Ш | Е/Ё  П  Щ | Ж  Р  Ы | З  С  Э | И  Т  Ю | Й  У  Я |
| Номер варианта | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Таблица 2

Индивидуальные задания

|  |  |
| --- | --- |
| № варианта | Задача |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |

**Рекомендации по выполнению задания**

Пример выполнения задания в [1, с. 59-63]:

<https://znanium.com/read?id=368962>

Задание выполнить и сохранить в одном файле с именем **Задание\_1\_Фамилия.docx** или **Задание\_1\_Фамилия.doc**.

# Бланк выполнения задания 1

1. Листинг программы для реализации возможного алгоритма решения задачи.

2. Словарь программы.

3. Оценка характеристик программы.

# Практическое задание 2

Тема «Оценка структурной сложности программ»

Задание. С целью оценки алгоритмической сложности необходимо:

* «разработать алгоритм решения задачи (см. табл. 2);
* построить граф потока управления;
* сформировать маршруты тестирования в соответствии с критериями 1, 2 и 3;
* определить значение цикломатического числа, характеризующего структурную сложность программ;
* сформировать матрицы смежности и достижимости;
* провести анализ полученных результатов, сформировав содержательные выводы» [1].

*Номер варианта выбирается из таблицы 1 по первой букве фамилии студента.*

Таблица 1

* Таблица выбора номера варианта

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Первая буква фамилии студента | А  К  Ф | Б  Л  Х | В  М  Ц | Г  Н  Ч | Д  О  Ш | Е/Ё  П  Щ | Ж  Р  Ы | З  С  Э | И  Т  Ю | Й  У  Я |
| Номер варианта | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Таблица 2

Индивидуальные задания

|  |  |
| --- | --- |
| № варианта | Задача |
| 1 | Вывести на экран сумму *S* ряда чисел: .  Сумма вычисляется до получения слагаемого, меньшего заданного значения *A*. Значение *A* ввести с клавиатуры. |
| 2 | Массив размерности *N* заполнен случайными целыми числами. Изменить порядок следования элементов в массиве на обратный. Значение *N* ввести с клавиатуры. |
| 3 | Вывести на экран все натуральные трехзначные числа из диапазона от *A* до *B*, значение которых кратно 17. При отсутствии чисел с указанными свойствами сформировать сообщение «Требуемых чисел нет». Границы диапазона *A* и *B* ввести с клавиатуры. |
| 4 | Массив размерности *N* заполнен случайными целыми числами. Упорядочить элементы массива по возрастанию модулей. Значение *N* ввести с клавиатуры. |
| 5 | Вывести на экран все натуральные числа из диапазона от *A* до *B*, сумма цифр которых равна *S*. При отсутствии чисел с указанными свойствами сформировать сообщение «Требуемых чисел нет». Границы диапазона *A* и *B* и заданная сумма цифр *S* вводятся с клавиатуры. |
| 6 | Массив заполнен случайными целыми числами на интервале [0, *K*]. Значение *K* ввести с клавиатуры. Подсчитать количество простых чисел в данном массиве. |
| 7 | Вывести на экран все натуральные числа из диапазона от *A* до *B*, в записи которых цифра 7 встречается ровно *N* раз. При отсутствии чисел с указанными свойствами сформировать сообщение «Требуемых чисел нет». Границы диапазона *A* и *B*, значение *N* ввести с клавиатуры. |
| 8 | Массив заполнен случайными целыми числами на интервале [0, *K*]. Значение *K* ввести с клавиатуры. Подсчитать количество чисел, кратных 3. |
| 9 | Массив размерности *N* заполнен случайными целыми числами. Упорядочить элементы массива по убыванию. Значение *N* ввести с клавиатуры. |
| 10 | Вычислить сумму *S* первых *N* слагаемых последовательности чисел, образуемых по правилу: .  Сумму вычислить двумя способами: S1 – суммирование от первого слагаемого до *N*-го слагаемого, S2 – суммирование от *N*-го слагаемого до первого слагаемого. Значение *N* ввести с клавиатуры. Вывести на экран вычисленные суммы S1 и S2, а также значение модуля разности между ними. |

**Рекомендации по выполнению задания**

Примеры выполнения задания в [1, с. 134-169]:

<https://znanium.com/read?id=368962>

Задание выполнить и сохранить в одном файле с именем **Задание\_2\_Фамилия.docx** или **Задание\_2\_Фамилия.doc**.

# Бланк выполнения задания 2

1. Блок-схема алгоритма решения задачи.

2. Управляющий граф задачи.

3. Оценка алгоритмической сложности.

# Практическое задание 3

Тема «Процедурно-ориентированные метрики»

Задание. Для оценки качества программы необходимо:

* разработать программу, реализующую предложенные в условии действия (см. табл. 2);
* вычислить количество функциональных указателей;
* оценить уровень связности и силу сцепления программных модулей;
* провести анализ полученных результатов.

*Номер варианта выбирается из таблицы 1 по первой букве фамилии студента.*

Таблица 1

* Таблица выбора номера варианта

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Первая буква фамилии студента | А  К  Ф | Б  Л  Х | В  М  Ц | Г  Н  Ч | Д  О  Ш | Е/Ё  П  Щ | Ж  Р  Ы | З  С  Э | И  Т  Ю | Й  У  Я |
| Номер варианта | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Таблица 2

Индивидуальные задания

|  |  |
| --- | --- |
| № варианта | Задача |
| 1 | Функция должна находить в массиве минимальный по модулю элемент и заменять им все элементы с нечётными номерами. |
| 2 | Функция должна вычислять сумму элементов массива с чётными номерами. |
| 3 | Функция должна в матрице находить сумму элементов *i*-й строки. |
| 4 | Функция должна переворачивать строку, передаваемую ей в качестве параметра, в зеркальное положение. |
| 5 | Функция должна вычислять сумму и количество элементов массива, меньших заданного числа *M*. |
| 6 | Функция должна вычислять произведение элементов массива с нечётными номерами. |
| 7 | Функция должна вычислять отношение количества положительных элементов массива к количеству отрицательных. |
| 8 | Дано натуральное число *N*. Вывести на экран число, которое получится после записи цифр числа *N* в обратном порядке. Для получения нового числа составить функцию. |
| 9 | Даны две квадратные матрицы. Определить матрицу, которая имеет минимальный «след», т.е. сумму элементов главной диагонали. Разработать функцию для нахождения «следа» матрицы и функцию вывода матрицы на экран. |
| 10 | Функция должна вычислять количество минимальных элементов в целочисленной матрице. |

**Рекомендации по выполнению задания**

Примеры выполнения задания в [1, с. 190-206]:

<https://znanium.com/read?id=368962>

Задание выполнить и сохранить в одном файле с именем **Задание\_3\_Фамилия.docx** или **Задание\_3\_Фамилия.doc**.

# Бланк выполнения задания 3

1. Текст программы для реализации возможного решения поставленной задачи.

2. Оценка характеристик программы.

# Практическое задание 4

Тема «Объектно-ориентированные метрики»

Задание. «Разработать приложение для вычисления результирующей информации об объектах, описанных с помощью наследования:

* базовый объект – «Тарифный план сотовой связи» (поле: стоимость минуты разговора – sm);
* производный объект 1 – «Телефон МТС» с полями: тип сотовой связи – Type, сумма на счете – schet, особые условия);
* производный объект 2 – «Телефон Мегафон» с полями: тип сотовой связи – Type, сумма на счете – schet, количество бесплатных минут – kolmin» [1].

Для выполнения задания необходимо:

а) «Определить базовый класс и производные классы, используя наследование.

б) Используя виртуальный метод печати базового класса, разработать переопределенные методы производных классов для вывода результирующей информации:

* тип сотовой связи;
* общая продолжительность разговора в рамках денежной суммы, хранящейся на счете.

в) Создать массив для хранения ссылок на объекты, следующие в произвольном порядке. Всю необходимую информацию вводит пользователь.

г) Создать объекты, присвоив начальные значения полям объекта с помощью конструктора.

д) Используя массив ссылок и цикл, вывести результирующую информацию» [1].

**Рекомендации по выполнению задания**

Необходимо разработать программу, реализующую алгоритм решения задачи, и оценить характеристики программы, используя объектно-ориентированные метрики Абреу.

Пример выполнения задания в [1, с. 267-270]:

<https://znanium.com/read?id=368962>

Задание выполнить и сохранить в одном файле с именем **Задание\_4\_Фамилия.docx** или **Задание\_4\_Фамилия.doc**.

# Бланк выполнения задания 4

1. Листинг программы, реализующей возможный алгоритм решения задачи.

2. Оценка характеристик программы. В соответствии с теорией Абреу необходимо определить значения следующих характеристик:

* «фактора закрытости метода (МНF);
* фактора закрытости свойства (AHF);
* фактора наследования метода (MIF);
* фактора наследования свойства (AIF);
* фактора полиморфизма (POF);
* фактора сцепления (СОF)» [1].

# Практическое задание 5

Тема «Оценка надежности программных средств»

Задание. Требуется провести расчет надежности программного обеспечения по результатам испытаний с помощью модели Джелински-Моранды (см. табл. 2).

*Номер варианта выбирается из таблицы 1 по первой букве фамилии студента.*

Таблица 1

Таблица выбора номера варианта

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Первая буква фамилии студента | А  К  Ф | Б  Л  Х | В  М  Ц | Г  Н  Ч | Д  О  Ш | Е/Ё  П  Щ | Ж  Р  Ы | З  С  Э | И  Т  Ю | Й  У  Я |
| Номер варианта | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Таблица 2

Индивидуальные задания

|  |  |
| --- | --- |
| № варианта | Задача |
| 1 – 4 | В результате тестирования программы серией из шести случайно выбранных из набора тестов обнаружено 2 ошибки. Ошибки обнаружены вторым и шестым тестами. Требуется определить количество ошибок *N* в программе до начала тестирования. |
| 5 – 7 | В результате тестирования программы серией из 25 случайно выбранных из набора тестов обнаружено 3 ошибки. Ошибки обнаружены четвертым, десятым и двадцать вторым тестами. Требуется определить количество ошибок *N* в программе до начала тестирования. |
| 8 – 10 | В результате тестирования программы серией из 11 случайно выбранных из набора тестов обнаружено 2 ошибки. Ошибки обнаружены четвертым и одиннадцатым тестами. Все ошибки исправлены сразу после обнаружения. В предположении, что исправление ошибок не повлекло появление новых ошибок, требуется оценить количество оставшихся в программе ошибок. |

**Рекомендации по выполнению задания**

Пример выполнения задания в [1, с. 287-289]:

<https://znanium.com/read?id=368962>

Задание выполнить и сохранить в одном файле с именем **Задание\_5\_Фамилия.docx** или **Задание\_5\_Фамилия.doc**.

# Бланк выполнения задания 5

1. Определение количества ошибок до начала тестирования программы.

2. Определение количества ошибок в программе, не устраненных после проведения тестирования.

# Практическое задание 6

Тема «Оценка характеристик при стандартизации и сертификации программных средств»

Задание. «В информационной системе используется пять типов запросов, обслуживаемых по модели с абсолютными приоритетами. Каждый из них запрашивает только один документ. Время поступления и длительности обработки запросов представлены в таблице 1.

Таблица 1

Исходные данные

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Запрос | *A* | *B* | *C* | *D* | *E* |
| Приоритет | 2 | 3 | 1 | 4 | 5 |
| Момент поступления , мин. | 0 | 4 | 5 | 6 | 7 |
| Время обслуживания , мин. | 2 | 3 | 6 | 1 | 2 |

Определить, удовлетворяет ли такая информационная система критерию своевременности представления запрашиваемой выходной информации, если предельное среднее время реакции системы на обработку запроса равно 7 минутам» [1].

Решить задачу с помощью модели бесприоритетного обслуживания.

**Рекомендации по выполнению задания**

Пример выполнения задания в [1, с. 381-384]:

<https://znanium.com/read?id=368962>

Задание выполнить и сохранить в одном файле с именем **Задание\_6\_Фамилия.docx** или **Задание\_6\_Фамилия.doc**.

# Бланк выполнения задания 6

1. Построение диаграммы поступления и исполнения запросов по исходным данным.

2. Определение среднего значения времени выполнения запросов.

3. Анализ результатов обслуживания запросов.

**Библиографический список**

1. Черников, Б. В. Оценка качества программного обеспечения: Практикум : учебное пособие / Б. В. Черников. Б. Е. Поклонов : под ред. Б. В. Черникова. – Москва : ИД «ФОРУМ» : ИНФРА-М, 2018. – 400 с. – URL: https://znanium.com/catalog/document?id=368962 (дата обращения: 12.02.2021).

2. Черников, Б. В. Управление качеством программного обеспечения : учебник / Б. В. Черников. – Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. – 240 с. URL: https://znanium.com/catalog/document?id=339309 (дата обращения: 12.02.2021).