

Паспорт курсовой работы

по дисциплине «Технологии и методы программирования», 4 семестр

1. Методика оценки.

Задание:

Цель работы: написать приложение для поиска в Интернете (начиная с заданного списка URL, см. п. 1 таблицы вариантов заданий, и далее по ссылкам вплоть до заданной в п. 2 глубины) документов, удовлетворяющих заданному условию (см. п. 3). Результаты поиска сформировать согласно п. 4 и сохранить (выдать на консоль) согласно п. 5. Пункт 6 определяет, нужно ли вести журналирование выполняемых операций в лог-файле. Приложение может быть одно или многопоточным и реализуется в соответствии с п. 7. Номер варианта задания в виде последовательности из 7 цифр, соответствующих содержанию второй колонки таблицы, выдается преподавателем.

Общие требования к выполнению работы:

Приложение может быть консольным или графическим, в случае реализации графического пользовательского интерфейса обработка командной строки заменяется обработкой текста, вводимого пользователем в единственное поле (разбиение вводимых данных на несколько полей приведет к уменьшению балла за разбор командной строки).

Приложение должно быть создано в одной из оболочек NetBeans, Eclipse или IntelliJIdea (либо с использованием утилит ant или maven). Варианты задания	Варианты	Балл
Пункт задания		

1. Начальный список URL

1.1. Передается в командной строке	5
1.2. Задан в файле, имя которого передается в командной строке	10

2. Глубина просмотра ссылок из начального списка URL

2.1. Нет	0
2.2. Константа, большая единицы	5
2.3. Задается в командной строке	10

3. Критерий отбора документов

3.1. Набор ключевых слов передается в командной строке	5
3.2. Набор ключевых слов с признаком, должно входить (+) или должно	10

отсутствовать (-) передается в командной строке

3.3.Набор ключевых слов задан в файле, 10
имя

Структура:

1. Введение;
2. Техническое задание.
 - основания для разработки;
 - назначение разработки;
 - требования к приложению.
3. Проектирование (разработка иерархии классов).
4. Структурное описание разработки (описание классов: спецификации полей, конструкторов, методов).
5. Функциональное описание разработки (описание алгоритмов).
6. Описание пользовательского интерфейса.
7. Тестирование программы.
8. Руководство пользователя.
9. Руководство программиста.(установка программы, необходимые программные и аппаратные средства)
10. Заключение
11. Список литературы.
12. Приложение (текст программы с комментариями).

2. Критерии оценки.

Работа считается выполненной на **пороговом** уровне, если приложение повторно просматривает ранее обработанные сайты, неверно вычисляет результаты обработки за счет гонок данных, или содержит другие серьезные ошибки, оценка составляет от 50 до 72 баллов.

Работа считается выполненной на **базовом** уровне, если приложение неверно определяет кодировку данных документа, или содержит подобные не очень серьезные ошибки, оценка составляет от 73 до 86 баллов.

Работа считается выполненной на **продвинутом** уровне, если приложение реализует все требуемые функции и имеет дружелюбный интерфейс, оценка составляет от 87 до 100 баллов

3. Шкала оценки.

В общей оценке по дисциплине баллы за работы учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Примерный перечень тем курсового проекта (работы).

Тема курсового проекта определяется способом задания списка URL: список, дерево и т.д..

5. Перечень вопросов к защите курсового проекта (работы).

1. Парадигмы программирования. Предшественники ООП. Основные принципы объектно-ориентированного программирования.
2. Язык программирования и платформа Java. Общая характеристика языка. Состав и характеристика компонент платформы.
3. Жизненный цикл приложений, написанных на Java. Состав и характеристика утилит, с помощью которых создаются приложения.
4. Типы данных языка Java. Сужающие и расширяющие преобразования. Области видимости объявлений.
5. Операции языка Java. Приоритеты операций. Вычисление выражений. Особенности целочисленной арифметики и арифметики чисел с плавающей точкой.
6. Операторы языка Java. Характеристика всех операторов. Особенности циклов, обработки исключений, синхронизации, проверки утверждений.
7. Структура Java-программы. Элементы соглашений о кодировании и именовании объектов. Понятие пакета. Директивы импорта.
8. Классы, заголовок класса. Объявление и инициализация полей класса. Модификаторы объявлений классов и их полей.
9. Конструкторы. Инициализаторы. Статические инициализаторы.
10. Объявления методов класса. Перегрузка методов. Понятия сигнатуры и контракта метода.
11. Создание и уничтожение объектов.
12. Наследование в Java. Затенение полей. Замещение методов. Доступность полей и методов классов.
13. Абстрактные классы, их назначение и использование. Объявления полей и методов абстрактных классов.
14. Интерфейсы, их объявление. Особенности наследования интерфейсов. Интерфейс-маркеры.
15. Статические внутренние классы. Особенности объявления и использования.
16. Нестатические внутренние классы. Особенности объявления и использования.
17. Локальные внутренние классы. Особенности объявления и использования.
18. Анонимные внутренние классы. Особенности объявления и использования.
19. Массивы в Java. Создание и инициализация. Ступенчатые массивы.
20. Базовые пакеты платформы Java. Пакет `java.lang`. Состав и общая характеристика. Класс `Object`.
21. Пакет `java.lang`. Классы `System` и `Runtime`. Характеристика наиболее употребительных методов.
22. Пакет `java.lang`. Работа со строками. Классы `String`, `StringBuffer` и `StringBuilder`.
23. Пакет `java.lang`. Класс `Enum`. Перечисления. Особенности перечислений в Java.
24. Пакет `java.lang`. Классы-обертки примитивных типов.
25. Пакет `java.lang`, класс `Math` и пакет `java.math`.
26. Пакет `java.net`. Основные принципы и протоколы работы в сети.
27. Пакет `java.net`. Классы `InetAddress`, `Socket` и `URL`.
28. Пакет `java.lang`. Потoki и многопоточность. Базовые механизмы. Класс `Thread` и интерфейс `Runnable`. Обычные потоки и потоки-демоны. Приоритеты потоков.
29. Классы `java.util.TimerTask` и `java.util.Timer`, их использование в многопоточных приложениях.

30. Многопоточность и проблемы, вызываемые неатомарностью операций: гонки данных. Модель памяти виртуальной машины Java.
31. Базовые средства Java для синхронизации потоков. Недостатки базовых средств синхронизации.
32. Пакет `java.util.concurrent`. Состав расширенных средств многопоточности. Интерфейс `Callable`.
33. Пакет `java.util.concurrent`. Фреймворк `Executor`. Интерфейсы `Executor`, `ExecutorService` и `ScheduledExecutorService`.
34. Пакет `java.util.concurrent`. Фреймворк `Executor`. Пулы потоков. Фабрика классов `Executors`.
35. Пакет `java.util.concurrent`. Расширения фреймворка `Executor` – `ForkJoinPool` и `ThreadPoolExecutor`.
36. Пакет `java.util.concurrent.locks`. Блокировки. Класс `ReentrantLock`.
37. Пакет `java.util.concurrent`. Синхронизаторы. Семафоры.
38. Пакет `java.util.concurrent`. Синхронизаторы. Барьеры.
39. Пакет `java.util.concurrent`. Синхронизаторы. Обменники.
40. Пакет `java.util.concurrent`. Синхронизаторы. Защелки.
41. Пакет `java.util.concurrent.atomic`. Классы с атомарными операциями.
42. Пакет `java.util`. Фреймворк коллекций. Основные интерфейсы.
43. Пакет `java.util`. Фреймворк коллекций. Интерфейсы `Collection` и `Set`. Классы-множества.
44. Пакет `java.util`. Фреймворк коллекций. Интерфейсы `List` и `ListIterator`. Классы-списки.
45. Пакет `java.util`. Фреймворк коллекций. Интерфейсы `Map` и `HashMap`. Классы-отображения.
46. Пакет `java.util`. Фреймворк коллекций. Интерфейсы `Queue` и `Deque`. Классы-очереди.
47. Пакеты `java.util` и `java.util.concurrent`. Классы «параллельных» коллекций.
48. Пакет `java.util`. Классы `Collections`, `BitSet` и `Arrays`.