графический интерфейс

Проект III

14 мая 2023 г.

* Описание Проекта

Первое в истории текстовое сообщение было отправлено 3 декабря 1993 года. С тех пор подавляющее большинство мобильных устройств имеют возможность отправлять сообщения такого типа. Результатом этого проекта станет приложение, имитирующее передачу сообщений от отправителя (мобильного устройства) через ряд сетевых слоев (промежуточных станций) к получателям (мобильным устройствам).

Используя среду SWING, создайте многопоточное оконное приложение, которое имитирует отправку SMS-сообщений от отправителей получателям. В визуальном слое приложение будет состоять из 3-х интерактивных панелей (объектов, унаследованных от класса JPanel), организованных с помощью автоматической разметки BorderLayout.

Крайняя левая панель будет посвящена отображению устройств вещания. Эта панель будет состоять из компонента JScrollPane, расположенного в центральной части, и кнопки JButton "Добавить", расположенной внизу панели. Внутри компонента JScroll-Pane будет размещена панель JPanel, которая с помощью вертикального BoxLayout расположит все компоненты, визуализирующие «виртуальные вещательные устройства» (VBD).

При нажатии кнопки «Добавить» в панели передающих устройств появится диалоговое окно, позволяющее ввести и подтвердить короткое текстовое сообщение. На основе введенного сообщения будет создан объект, реализующий виртуальное вещательное устройство (ВВУ), экземпляры которого будут визуализированы в панели.

Визуализация объекта VBD будет основана на компоненте JPanel и будет состоять из компонентов, описанных JLabel:

* JSlider — указание частоты отправки сообщений;
* JButton – позволяет прекратить работу VBD (что приведет к удалению устройства);
* JTextField – нередактируемое поле, представляющее номер устройства;
* JComboBox — позволяющий указать состояние WAITING или ACTIVE, на основании которого поток, выполняющий данный VBD, будет находиться в приостановленном или рабочем состоянии;

Крайняя правая панель будет посвящена отображению принимающих устройств. Структура этой панели будет аналогична представленной структуре панели передающих устройств. Однако он покажет виртуальные приемные устройства (VRD), объекты которых будут созданы в результате нажатия кнопки «Добавить».

Визуализация объекта VRD будет состоять из компонентов, описанных JLabel:

* JButton — позволяющая прекратить работу VRD (что приведет к удалению устройства);
* JLabel — представление информации о количестве полученных сообщений;
* JCheckBox — указывает, должна ли информация о количестве полученных сообщений удаляться каждые 10 секунд, если отмечено или нет.

1 из 3

графический интерфейс

Проект III

14 мая 2023 г.

Средняя панель позволит вам визуализировать базовые станции (BTS) на внешних слоях и станции контроллеров (BSC) на средних слоях. Эта панель будет состоять из компонента JPanel, расположенного в центральной части, содержащего горизонтально ориентированные визуализации слоев BTS и BCS, и кнопок добавления и удаления слоев контроллеров (BSC), расположенных в нижней части компонента.

Каждый уровень BTS или BSC реализуется JPanel, в котором визуальные компоненты, представляющие одну станцию (BTS или BCS), расположены вертикально.

Каждая станция BTS или BSC представлена в виде компонента JPanel, состоящего из следующих компонентов:

* JLabel – показывает другой, уникальный и неслучайный номер станции;
* JLabel – количество обработанных SMS-сообщений;
* JLabel – количество SMS, ожидающих отправки;
* Работа приложения

Пользователь приложения создает ряд VBD-объектов, каждый из которых представляет собой отдельный поток, и сразу же начинает передавать введенное при создании сообщение. Поскольку, согласно стандарту, каждое SMS имеет номер отправителя и получателя, закодированный вместе с сообщением, в качестве получателя выбирается случайный элемент VRD. Созданное SMS пересылается на станцию BTS с наименьшим количеством SMS, ожидающих отправки.

Приложение среды выполнения содержит три уровня. Между входным и выходным уровнями должен быть хотя бы один средний уровень BSC. Точное количество промежуточных слоев зависит от действий пользователя приложения, который может добавить или удалить слой с помощью клавиш. Каждый вновь созданный коммуникационный уровень будет создан с одним BSC, и удаление уровня приведет к тому, что этот уровень перестанет принимать сообщения и немедленно перенаправит сообщения от всех BSC, пропустив время передачи.

Пересылка образца SMS будет выглядеть так:

VBD →BTS →BSC →···→BTS →VRD

Следует предположить, что:

* каждый BSC будет хранить SMS в течение произвольного времени (от 5 до 15 секунд), а затем пересылать его на следующий уровень;
* каждая BTS пересылает SMS на следующий уровень или VRD через 3 секунды;
* при пересылке SMS на следующий уровень всегда выбирается тот BTS/BSC, который содержит наименьшее количество SMS;
* если на данном уровне количество SMS в каждом BSC или BTS больше 5, автоматически добавляется новый BTS/BSC;

2 из 3

графический интерфейс

Проект III

14 мая 2023 г.

Если выяснится, что номер получателя, на который было отправлено SMS, не существует, программа выдаст исключение, и система обработает его, выведя сообщение в консоль.

Предположим, что:

* сообщения отправляются в виде PDU, определенного в стандартах Европейского института телекоммуникационных стандартов (ETSI), описанных в документах GSM 03.38, GSM 03.40, GSM 03.41, при этом прилагаемый документ с описанием кодирования является лишь выдержкой из этих стандартов;
* при закрытии приложения будет создан бинарный файл, представляющий информацию о каждом VBD, содержащий информацию о количестве отправленных сообщений и содержащий отправленное сообщение.
* Формальные требования

Решение в виде поставляемого приложения должно быть представлено до 28 мая 2023 года на платформе Teams.

Проект требует использования всех вопросов, обсуждавшихся на лекциях и упражнениях, в том числе, например. объектно-ориентированное программирование, коллекции, исключения, потоки, ввод/вывод и многое другое. При этом нельзя использовать никакие внешние библиотеки, чтобы гарантировать, что весь код основан только на стандартной библиотеке Java.

Чтобы обеспечить правильное и безопасное выполнение параллельных операций, необходимо реализовать их с помощью потоков (Thread или Runnable). Использование других инструментов, таких как Timer или Executor, не допускается.

Для правильной реализации проекта необходимо предоставить две версии программы:

* один файл с именем SxxxxxProjekt03.java, который будет содержать весь исходный код программы, включая внешние классы. Этот файл должен компилироваться без ошибок;
* сжатый архив с именем SxxxxxProjekt03.zip, содержащий все файлы проекта, включая исходные файлы.

Обратите внимание, что файл SxxxxxProject03.java должен быть автономным, содержать все необходимые внешние классы и компилироваться без ошибок. Архив, в свою очередь, должен содержать весь проект, включая исходные файлы.

Примечание: несоблюдение формальных требований влечет за собой умножение полученной оценки на коэффициент 0,3.

3 из 3