Разработка автоматизированного мониторинга веб-сайтов на C++

Техническое задание

Листов 14

Пермь 2025

**Оглавление**

[1. Введение 3](#_heading=h.1fob9te)

[2. Основания для разработки АРМ 3](#_heading=h.3znysh7)

[3. Назначение АРМ 3](#_heading=h.2et92p0)

4.[Требования к](#_heading=h.3dy6vkm) АРМ 6

5. [Требования к программной документации](#_heading=h.1t3h5sf) 11

6. [Стадии и этапы разработки 1](#_heading=h.2s8eyo1)1

7. UML-диаграммы 13

8. Библиотеки, API и инструменты 14

9. [Порядок контроля и приемки 1](#_heading=h.17dp8vu)7

Список литературы 17

**1. Введение**

**1.1. Наименование программы**

Наименование программы: “ Разработка автоматизированного рабочего места специалиста по мониторингу курсов валют на веб-сайте на C++”. Далее по тексту – АРМ.

**1.2. Краткая характеристика области применения**

АРМ , предназначен для мониторинга курсов валют на заданном веб-сайте. АРМ реализовано на языке C++ с использованием фреймворка Qt, что обеспечивает кроссплатформенность и удобный графический интерфейс пользователя (GUI). Основная цель разработки - предоставить пользователю удобный инструмент для отслеживания изменений курсов валют в режиме реального времени с возможностью визуализации этих изменений.

**2. Основания для разработки АРМ**

**2.1. Основания для проведения разработки**

“ Разработка автоматизированного рабочего места специалиста по управлению котельной на C++” разрабатывается в соответствии с первой частью творческого задания по разработке автоматизированного рабочего места специалиста. Задание утверждено доцентом кафедры ИТАС Поляковой О. А. от 06.05.2025.

Согласно требованиям Поляковой О. А., именуемой заказчиком, исполнители: Дурбажев Д.И., Черемных Н.П., обязаны разработать систему не позднее 17.05.2025.

**3. Назначение АРМ**

**3.1. Функциональное назначение**

АРМ предназначено для выполнения следующих функций:  
  
• **Автоматический сбор данных:** Периодический автоматический запрос данных о курсах валют с указанного веб-сайта.  
• **Обработка данных:** Извлечение значений курсов валют из полученного HTML-кода веб-страницы.  
• **Отображение данных:** Представление текущих значений курсов валют в графическом интерфейсе пользователя.  
• **Визуализация изменений:** Отображение изменений курсов валют с помощью визуальных индикаторов (например, изменение цвета фона).  
• **Управление мониторингом:** Предоставление пользователю возможности запускать и останавливать процесс мониторинга.  
• **Предоставление ссылок на источники данных:** Обеспечение быстрого доступа к веб-сайтам с данными о курсах валют

**3.2. Эксплуатационное назначение**

Автоматизированное рабочее место "WebMonitor" предназначено для использования широким кругом пользователей, преследующих различные цели, связанные с отслеживанием курсов валют. Детальное описание эксплуатационного назначения по категориям пользователей представлено ниже:  
  
• **Финансовые аналитики:** АРМ "WebMonitor" позволяет финансовым аналитикам в режиме реального времени отслеживать динамику курсов валют, что необходимо для проведения фундаментального анализа и прогнозирования трендов на финансовых рынках. Аналитики могут использовать полученные данные для оценки влияния изменений валютных курсов на финансовые показатели компаний, отраслей и экономики в целом. Оперативное получение информации позволяет своевременно корректировать стратегии и минимизировать риски.  
  
• **Трейдеры:** Для трейдеров, занимающихся краткосрочной торговлей на валютном рынке (Forex), "WebMonitor" является незаменимым инструментом для мониторинга ключевых валютных пар. Быстрое и наглядное отображение изменений курсов позволяет оперативно реагировать на колебания рынка и принимать решения о покупке или продаже валюты. Индикаторы изменений, визуализируемые АРМ, помогают выявлять моменты для входа и выхода из сделок.  
  
• **Инвесторы:** Инвесторы, вкладывающие средства в активы, номинированные в различных валютах (например, акции иностранных компаний, облигации, недвижимость), могут использовать "WebMonitor" для контроля за стоимостью своих инвестиций. Изменения валютных курсов оказывают существенное влияние на доходность инвестиций, поэтому своевременное отслеживание этих изменений позволяет инвесторам принимать обоснованные решения о ребалансировке портфеля и хеджировании валютных рисков.  
  
• **Предприятия, ведущие внешнеэкономическую деятельность:** Компании, осуществляющие экспортно-импортные операции, нуждаются в постоянном мониторинге курсов валют для определения оптимальных условий заключения контрактов, расчета себестоимости продукции и управления валютными рисками. "WebMonitor" предоставляет возможность оперативно отслеживать изменения курсов валют, что позволяет компаниям принимать взвешенные решения и повышать эффективность внешнеэкономической деятельности.  
  
• **Бухгалтеры и финансовые менеджеры:** Для правильного учета финансовых операций, проводимых в иностранной валюте, необходимо знать текущие курсы валют. АРМ "WebMonitor" позволяет получать актуальную информацию о курсах валют, необходимую для бухгалтерского учета и финансового планирования.  
  
• **Студенты и исследователи:** "WebMonitor" может быть использован в образовательных целях для изучения финансовых рынков и разработки алгоритмов анализа данных. Студенты могут использовать АРМ для проведения исследований, моделирования различных сценариев и анализа влияния изменений валютных курсов на экономические процессы.  
  
• **Любые пользователи, заинтересованные в отслеживании курсов валют:** "WebMonitor" может быть полезен для частных лиц, планирующих путешествия за границу, совершающих покупки в интернет-магазинах, работающих с иностранными партнерами или просто интересующихся экономической ситуацией в мире. Например, при планировании поездки можно отслеживать курс валюты страны назначения, чтобы выбрать наиболее выгодный момент для обмена валюты.

**4. Требования к АРМ**

**4.1. Требования к функциональным характеристикам**

**4.1.1. Требования к составу выполняемых функций**

• **Мониторинг курсов валют:**  
  • АРМ должен обеспечивать автоматический и периодический мониторинг курсов валют по списку: USDRUB, EURUSD, JPYUSD, ZARUSD.  
  • Должна быть предусмотрена возможность (желательно) добавления/удаления валютных пар из списка мониторинга (в идеале - через конфигурационный файл или GUI).  
  • АРМ должен осуществлять мониторинг курсов валют, получая данные с предопределенного веб-сайта ([finance.yahoo.com](https://finance.yahoo.com/" \o "https://finance.yahoo.com/" \t "https://web.telegram.org/a/_blank)).  
  • Должна быть предусмотрена возможность (желательно) изменения URL-адреса веб-сайта, используемого для мониторинга, без перекомпиляции программы. Это может быть реализовано через конфигурационный файл или графический интерфейс.  
• **Получение данных:**  
  • АРМ должен автоматически отправлять HTTP-запросы к указанному веб-сайту.  
  • АРМ должен корректно обрабатывать HTTP-ответы, включая случаи ошибок (например, HTTP 404, HTTP 500).  
  • АРМ должен извлекать данные о курсах валют из полученного HTML-кода веб-страницы. Для этого должны использоваться методы парсинга HTML (например, с использованием QDomDocument или QRegularExpression).  
• **Отображение данных:**  
  • АРМ должен отображать текущие значения курсов валют в главном окне приложения.  
  • Значения курсов валют должны быть представлены в удобном для восприятия формате (например, с указанием валютной пары и значения).  
  • АРМ должен обеспечивать автоматическое обновление значений курсов валют через заданный интервал времени.  
• **Визуализация изменений:**  
  • АРМ должен визуально отображать изменения курсов валют.  
  • Визуализация должна осуществляться с помощью цветовых индикаторов (изменение цвета фона элемента интерфейса).  
  • Должна быть предусмотрена возможность настройки пороговых значений, определяющих изменение цвета индикатора. Например, если курс валюты вырос более чем на X%, цвет индикатора становится зеленым, а если упал более чем на Y%, цвет индикатора становится красным. Настройка порогов должна быть доступна пользователю (через конфигурационный файл или графический интерфейс).  
• **Управление мониторингом:**  
  • АРМ должен предоставлять пользователю возможность запуска и остановки процесса мониторинга. Это должно быть реализовано с помощью кнопки "Старт/Стоп".  
• **Предоставление ссылок:**  
  • АРМ должен предоставлять быстрый доступ к веб-сайтам, содержащим информацию о курсах валют (например, [finance.yahoo.com](https://finance.yahoo.com/" \o "https://finance.yahoo.com/" \t "https://web.telegram.org/a/_blank), [finam.ru](https://finam.ru/" \o "https://finam.ru/" \t "https://web.telegram.org/a/_blank)). Должны быть предусмотрены кнопки для быстрого перехода на эти веб-сайты.  
• **Обработка ошибок и исключений:**  
  • АРМ должен корректно обрабатывать возможные ошибки и исключения, возникающие в процессе работы (например, отсутствие интернет-соединения, некорректный формат данных на веб-сайте, ошибки при парсинге HTML).  
  • При возникновении ошибок АРМ должен отображать информативные сообщения для пользователя, позволяющие понять причину проблемы.  
  • АРМ не должен аварийно завершаться при возникновении ошибок.

**4.1.2. Требование к организации входных данных**

• **URL веб-сайта для мониторинга:**  
  • Должен быть предусмотрен способ задания URL веб-сайта, с которого АРМ получает данные о курсах валют. URL должен быть задан в формате строки (QString в Qt).  
  • (Желательно) URL должен быть настраиваемым, чтобы пользователь мог изменить веб-сайт для мониторинга.  
• **Интервал обновления данных:**  
  • Должен быть предусмотрен способ задания интервала времени, через который АРМ автоматически обновляет данные о курсах валют. Интервал должен быть задан в миллисекундах (int в Qt).  
  • (Желательно) Интервал обновления должен быть настраиваемым, чтобы пользователь мог изменить частоту обновления данных.  
• **Пороговые значения для индикаторов изменений:**  
  • Должен быть предусмотрен способ задания пороговых значений, определяющих изменение цвета индикаторов.  
  • Значения должны быть заданы в процентах (double в Qt).  
  • Должно быть предусмотрено два пороговых значения: одно для роста курса валюты, другое - для падения.  
  • (Желательно) Пороговые значения должны быть настраиваемыми для каждой валютной пары отдельно.  
• **(Опционально) Список валютных пар для мониторинга:**  
  • В случае реализации возможности добавления/удаления валютных пар, должен быть предусмотрен способ задания списка валютных пар. Список может быть задан в виде массива строк (QStringList в Qt).

**4.1.3. Требования к организации выходных данных**

• **Значения курсов валют:**  
  • АРМ должен отображать текущие значения курсов валют в главном окне приложения.  
  • Значения должны быть представлены в виде строк (QString в Qt).  
  • Формат представления значений должен быть четким и понятным для пользователя (например, "USDRUB: 75.50").  
• **Визуальные индикаторы изменений:**  
  • АРМ должен отображать визуальные индикаторы изменений курсов валют.  
  • Индикаторы должны представлять собой элементы графического интерфейса (например, QLabel с измененным цветом фона).  
  • Цвет индикатора должен соответствовать изменению курса валюты (например, зеленый - рост, красный - падение, серый - без изменений).  
• **Сообщения об ошибках:**  
  • При возникновении ошибок АРМ должен отображать сообщения об ошибках в понятном для пользователя формате.  
  • Сообщения об ошибках должны содержать информацию о причине возникновения проблемы.

**4.1.4. Требования к временным характеристикам**

• **Время запуска АРМ:** АРМ должен запускаться не более чем за 5 секунд.  
• **Время обновления данных:** Время, затрачиваемое на обновление данных о курсах валют (отправка HTTP-запроса, получение и парсинг ответа), не должно превышать 2 секунд.  
• **Интервал обновления данных:** АРМ должен поддерживать возможность задания интервала обновления данных, например 10000 миллисекунд, т.е. 10 секунд.  
• **Реакция на действия пользователя:** АРМ должен быстро реагировать на действия пользователя (например, нажатие кнопки "Старт/Стоп"). Время реакции не должно превышать 200 миллисекунд.

**4.2. Требования к надежности**

• **Устойчивость к ошибкам:** АРМ должен корректно обрабатывать ошибки и исключения, возникающие в процессе работы, и не приводить к аварийному завершению приложения.  
• **Восстановление после сбоев:** В случае временного отсутствия интернет-соединения или других сбоев АРМ должен автоматически восстанавливать свою работу после устранения проблемы.  
• **Защита от некорректных данных:** АРМ должен проверять корректность получаемых данных (например, проверять, что значения курсов валют являются числами) и не допускать отображения некорректной информации.  
• **Стабильность работы:** АРМ должен стабильно работать в течение длительного времени без сбоев и зависаний. Необходимо провести тестирование на длительность работы (например, 24 часа) для выявления возможных проблем.  
• **Корректное завершение работы:** АРМ должен корректно завершать свою работу при закрытии пользователем, освобождая все выделенные ресурсы (память, сетевые соединения).

**5. Требования к программной документации**

**5.1. Предварительный состав программной документации**

Состав программной документации должен включать в себя:

1. Техническое задание.

2. Общая документация по разработке

**6. Стадии и этапы разработки**

**Стадия 1: Планирование и анализ требований**  
  
• **Этап 1.1: Определение целей и задач проекта.** На этом этапе определяются цели разработки АРМ, его функциональное и эксплуатационное назначение, а также целевая аудитория.  
• **Этап 1.2: Сбор и анализ требований.** На этом этапе собираются и анализируются требования к АРМ, как функциональные, так и нефункциональные. Определяются источники данных, формат входных и выходных данных, требования к производительности, надежности и удобству использования.  
• **Этап 1.3: Разработка технического задания (ТЗ).** На основе анализа требований разрабатывается ТЗ, содержащее подробное описание АРМ, его функциональности, требований и критериев приемки.  
  
**Стадия 2: Проектирование**  
  
• **Этап 2.1: Разработка архитектуры АРМ.** На этом этапе определяется архитектура АРМ, включая структуру классов, взаимодействие между модулями и используемые библиотеки. Разрабатываются UML-диаграммы для визуализации архитектуры.  
• **Этап 2.2: Проектирование базы данных (если требуется).** В данном проекте база данных, скорее всего, не требуется, но если бы данные о курсах валют сохранялись для анализа, на этом этапе проектировалась бы структура базы данных.

• **Этап 2.3: Проектирование пользовательского интерфейса (UI).** На этом этапе разрабатывается дизайн UI с использованием Qt Designer. Определяются элементы управления, их расположение и взаимодействие.

• **Этап 2.4: Разработка спецификаций интерфейсов.** Определяются спецификации интерфейсов между модулями АРМ.

**Стадия 3: Реализация (Кодирование)**

• **Этап 3.1: Реализация модулей АРМ.** На этом этапе разрабатываются отдельные модули АРМ в соответствии с разработанной архитектурой и спецификациями. Используется язык C++ и фреймворк Qt.

• **Этап 3.2: Интеграция модулей.** На этом этапе отдельные модули интегрируются в единую систему.

• **Этап 3.3: Разработка пользовательского интерфейса.** На этом этапе создается UI на основе разработанного дизайна (с использованием Qt Designer).

• **Этап 3.4: Написание документации.** На этом этапе разрабатывается программная документация, включая руководство пользователя и руководство разработчика.

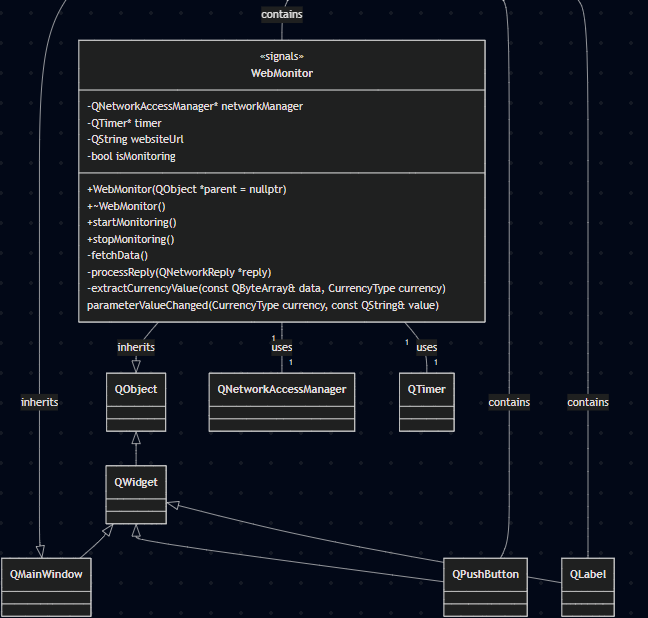
**Стадия 4: Тестирование**

• **Этап 4.1: Модульное тестирование.** На этом этапе проводится тестирование отдельных модулей АРМ для проверки их корректности и соответствия требованиям.

• **Этап 4.2: Интеграционное тестирование.** На этом этапе проводится тестирование взаимодействия между модулями АРМ.

• **Этап 4.3: Системное тестирование.** На этом этапе проводится тестирование АРМ как единой системы для проверки соответствия всем требованиям ТЗ.

1. **UML-диаграммы**   
   



1. **Библиотеки, API и инструменты**
   1. ****Основные библиотеки:****

• **Qt Framework:**  Qt 6.9.0

• QtCore: QObject.

• QtGui: QPalette, QColor.

• QtWidgets: QPushButton, QLabel.

• QtNetwork: QNetworkRequest, QNetworkReply.

• QtXml

• **C++ Standard Library:**

• Строки (std::string, QString).

• Контейнеры (QByteArray).

* 1. ****API:****

• **Qt Network API:**

• QNetworkAccessManager::get(): Для выполнения HTTP GET запросов.

• QNetworkRequest: Для создания HTTP запроса.

• QNetworkReply: Для получения ответа на HTTP запрос. reply->readAll(): Для чтения данных из ответа.

• QNetworkReply::error(): Для проверки наличия ошибок при выполнении запроса.

• **Qt GUI API:**

• QLabel::setText(): Для установки текста лейбла.

• QPushButton::setText(): Для установки текста кнопки.

• connect(): Для соединения сигналов и слотов.

• QTimer: Для периодического выполнения задачи (мониторинга). QTimer::start(), QTimer::stop().

• **Qt Core API:**

• QObject::connect(): Для соединения сигналов и слотов.

• QString, QByteArray: Классы для работы со строками и массивами байт. QString::toDouble(): Преобразование строки в число. QString::fromUtf8(): Преобразование данных из UTF-8 кодировки в QString.

• QUrl: Класс для работы с URL.

• QDesktopServices::openUrl(): Открытие URL в браузере.

• **Qt XML API**

• **QRegularExpression API:**

• QRegularExpression::match: Для поиска соответствия регулярному выражению в строке.

• QRegularExpressionMatch::hasMatch: Проверка успешности поиска.

• QRegularExpressionMatch::captured: Получение захваченной группы из соответствия.

* 1. **Инструменты разработки:**

• **Qt Creator:** Интегрированная среда разработки (IDE) для Qt. Используется для написания, отладки и сборки проекта. Файлы .pro и .pro.user создаются Qt Creator.

• **Qt Designer:** Инструмент для визуального создания GUI (файлы .ui). Генерирует C++ код, который описывает структуру интерфейса.

• **Компилятор C++:** MSVC (Windows). Используется для компиляции исходного кода в исполняемый файл.

• **Система сборки qmake:** используется для управления процессом сборки проекта.

• **Отладчик:** Встроенный в Qt Creator. Используется для поиска и исправления ошибок в коде.

* 1. **Внешние сервисы (в runtime):**

•**Веб-сайт с курсами валют (**[finance.yahoo.com/markets/currencies/](https://finance.yahoo.com/markets/currencies/" \o "https://finance.yahoo.com/markets/currencies/" \t "https://web.telegram.org/a/_blank)**):** Источник данных для мониторинга. Приложение делает HTTP запросы к этому сайту, чтобы получить текущие значения валют.

**9. Порядок контроля и приемки**

Приемосдаточные испытания системы должны проводиться на кафедре ИТАС ПНИПУ в сроки с 17.05.2025 до указанных заказчиком.

**Список литературы**

1. **Qt Documentation**. - URL: [doc.qt.io](https://doc.qt.io/" \o "https://doc.qt.io/" \t "https://web.telegram.org/a/_blank) (дата обращения: [16.05.2025]). - Текст : электрон. дан.
2. **Yahoo Finance**. Markets/Currencies. - URL: [https://finance.yahoo.com/markets/currencies/](https://finance.yahoo.com/markets/currencies/" \o "https://finance.yahoo.com/markets/currencies/" \t "https://web.telegram.org/a/_blank) (дата обращения: [16.05.2025]). - Текст : электрон. дан.
3. **Финам**. - URL: <https://www.finam.ru> (дата обращения: [16.05.2025]). - Текст : электрон. дан.