Тип работы: Реферат

Предмет: Программирование

Оглавление
Введение 2
Раздел 1. Система ООП JAVA. 3
Тема 1.1 Система ООП JAVA. 3
Синтаксис языка 3
Объекты и классы 4
Числа, строки и даты 5
Массивы и коллекции 6
Наследование и полиморфизм 7
Особенности ООП в Java 8
Исключения, отладка, тестирование и логирование 10
Работа с файлами и сетью 12
Работа с MySQL в Java 13
Многопоточность 14
Разработка веб-приложений 15
Нереляционные (NoSQL) базы данных 16
Производительность и оптимизация 16
Распределённые хранилища и вычисления 17
Раздел 2. Система ООП C++. 18
Тема 2.1 Система ООП C++. 18
Основы C++ 18
Работа с данными 19
Составные типы 20
Циклы и выражения отношений 21
Операторы ветвления и логические операции 22
Функции как программные модули C++ 23
Дополнительные сведения о функциях 23
Модели памяти и пространства имен 24
Объекты и классы 24
Работа с классами 25
Классы и динамическое выделение памяти 26
Наследование классов 27
Повторное использование кода в C++ 28
Класс string и стандартная библиотека шаблонов 28
Ввод, вывод и файлы 29
Раздел 3. Основы языка C#. 30
Тема 3.1 Основы языка C#. 30
Среда разработки 30
Синтаксис, Директивы, Классы, Методы 31
Типы данных, Переменные, Тип string 32
Конвертация строк и чисел 33
if, else if, else, DEBUG 33
Тернарная операция 35
MessageBox 35
Методы 36
Условные выражения 36
i++, i-- , +=, -=, инкремент, декремент 37

switch, case, break, default (условные конструкции) 37
try, catch, finally, Обработка исключений, throw new Exception ex 38
Циклы, while, do while 38
Операторы async await 39
Цикл for 40
Массивы, array 41
Цикл foreach 42
break, continue 42
goto и return 43
Раздел 4. Язык программирования PHP. 44
Тема 4.1 Язык программирования PHP. 44
Выражения и переменные, типы 44
Функции. Операторы контроля. Включение файлов 45
Массивы. Циклы. Взаимодействие с пользователем 46
Работа с файлами на сервере. Загрузка от клиента 47
Cookie. Сессии. Авторизация 48
Классы и объекты. Введение в ООП 48
Объектно-ориентированный подход 49
Работа с базой данных 50
Архитектура проекта 51
Модели данных и ООП 52
Изоляция уровня представления 53
Контроллеры и фронт-контроллер 54
Исключения 54
Современные стандарты PHP 55
Новые возможности PHP 56
Обзор современных фреймворков 56
Заключение 58
Список используемой литературы 59

Введение

Разработка программных модулей является фундаментальным этапом в создании компьютерных систем, обеспечивая их функциональность и расширяемость. В данном контексте, мы рассмотрим разработку программных модулей, ориентированных на различные языки программирования, такие как Java, C++, C# и PHP.

Системы объектно-ориентированного программирования (ООП) играют ключевую роль в этом процессе, обеспечивая структурирование кода, повторное использование компонентов и более высокий уровень абстракции. В Java и C++, ООП является неотъемлемой частью разработки, позволяя создавать классы, объекты и наследование для организации кода.

Основы языка C# открывают перед разработчиками мощные инструменты для создания модулей, включая поддержку ООП, управление памятью и множество библиотек для работы с разнообразными задачами.

В свою очередь, язык программирования PHP предназначен для разработки веб-приложений и обладает особенностями, направленными на работу с веб-серверами и базами данных.

В данном исследовании мы рассмотрим ключевые аспекты разработки программных модулей, сфокусировавшись на использовании систем ООП в Java и C++, а также рассмотрим основы языка C# и особенности PHP в контексте создания модулей для компьютерных систем.

Раздел 1. Система ООП JAVA.

Тема 1.1 Система ООП JAVA.

Синтаксис языка

Java является объектно-ориентированным языком. Это означает, что писать программы на Java нужно с применением объектно-ориентированного стиля. И стиль этот основан на использовании в программе объектов и классов.

Синтаксис языка Java - это набор правил и структур, которые определяют, как писать программы на этом языке. Вот ключевые элементы синтаксиса Java:
• Классы и Объекты: Java является объектно-ориентированным языком, где программа строится вокруг классов и объектов. Классы определяют шаблоны объектов, а объекты - их экземпляры.
• Переменные и Типы данных: Java поддерживает различные типы данных, такие как целые числа, числа с плавающей запятой, символы и булевы значения. Переменные используются для хранения данных.
• Операторы: Java включает в себя операторы для выполнения математических операций, сравнения и логических действий.
• Условные операторы: Для принятия решений в программах используются условные операторы, такие как if, else if и switch.
• Циклы: Java предоставляет циклы, такие как for, while и do-while, для выполнения повторяющихся действий.
• Массивы: Массивы позволяют хранить множество элементов одного типа данных в одной переменной.
• Методы: Методы - это функции в Java, которые выполняют определенные действия. Они могут быть вызваны из других частей программы.
• Исключения: Для обработки ошибок и исключительных ситуаций в Java используются блоки try, catch и finally.
• Наследование и Полиморфизм: Java поддерживает наследование, где один класс может наследовать свойства и методы от другого класса. Это также связано с полиморфизмом.
• Интерфейсы и Абстрактные классы: Java позволяет создавать интерфейсы и абстрактные классы для определения общих методов и свойств, которые должны быть реализованы в дочерних классах.
• Обработка строк: Java предоставляет множество операций для работы со строками, включая конкатенацию, поиск и изменение.
Все эти элементы синтаксиса Java являются основными для написания программ.
Объекты и классы
Класс – это шаблон для объекта. Он определяет, как объект будет выглядеть и какими функциями обладать. Каждый объект является объектом какого-то класса.
Метод класса - это блок кода, состоящий из ряда инструкций, который можно вызывать по его имени. Он обязательно содержит возвращаемый тип, название, аргументы и тело метода.
Шаблоном или описанием объекта является класс, а объект представляет экземпляр этого класса. Можно еще провести следующую аналогию. У нас у всех есть некоторое представление о человеке - наличие двух рук, двух ног, головы, туловища и т.д. Есть некоторый шаблон - этот шаблон можно назвать классом. Реально же существующий человек (фактически экземпляр данного класса) является объектом этого класса.
Класс определяется с помощью ключевого слова class. Для хранения состояния объекта в классе применяются поля или переменные класса. Для определения поведения объекта в классе применяются методы.
Как правило, классы определяются в разных файлах. В данном случае для простоты мы определяем два класса в одном файле. Стоит отметить, что в этом случае только один класс может иметь модификатор public, а сам файл кода должен называться по имени этого класса (Рисунок 1).

Рисунок 1 “Пример взаимодействия объектов и классов”
Числа, строки и даты
Java предоставляет различные типы данных для работы с числами (Рисунок 2), включая:
1. Целые числа:
• byte: 8-битное знаковое целое число.
• short: 16-битное знаковое целое число.
• int: 32-битное знаковое целое число (наиболее распространенный).
• long: 64-битное знаковое целое число.
2. Десятичные числа:
• float: 32-битное число с плавающей запятой.
• double: 64-битное число с плавающей запятой.
Например:
```
int integerNumber = 42;
double doubleNumber = 3.14;
```

Строки в Java представлены классом String. Также можно создавать строки и выполнять различные операции с ними, такие как конкатенация и поиск подстрок.
Например:
String hello = "Привет!";
String concatResult = hello + " Как дела?";
int stringLength = hello.length();
Java предоставляет классы для работы с датами и временем в пакете java.time. Наиболее важные классы включают в себя:
• LocalDate: Представляет дату.
• LocalTime: Представляет время.
• LocalDateTime: Представляет комбинацию даты и времени.
• DateTimeFormatter: Используется для форматирования и разбора даты и времени.

Рисунок 2 “Структура типов данных в Java”
В Java, работа с числами, строками и датами представляет собой фундаментальные аспекты программирования. Выбор подходящих типов данных для чисел и умение выполнять операции с ними критически важны. Строки позволяют манипулировать текстовой информацией, а работа с датами и временем, облегчает управление временными данными и форматирование.
Массивы и коллекции
В языке Java есть два типа конструкций, предназначенных для хранения нескольких значений или объектов одного типа: массивы и коллекции (Рисунок 3).
Массивы - это простые конструкции фиксированного размера, и поэтому они могут хранить только заданное количество элементов. Массивы встроены в ядро языка Java, и используемый при работе с ними синтаксис Java очень прост и понятен. Например, чтобы получить элемент массива с номером n, вам нужно вызвать функцию array[n]. Коллекции - это более сложный, но в то же время более гибкий тип данных. Прежде всего, размер коллекции можно изменять: вы можете добавлять в коллекцию любое количество элементов. Коллекции автоматически обрабатывают удаление элемента из любой позиции. В языке Java существует несколько типов коллекций с различной внутренней структурой хранения (линейная, список, хэш-набор, дерево элементов и другие). Вы можете выбрать тот тип коллекции, который лучше подходит для вашей задачи, чтобы наиболее часто используемые вами операции выполнялись наиболее эффективно. Коллекции - это Java классы, и для получения, например, nого элемента из коллекции collection типа ArrayList нужно будет вызвать метод collection.get(n).
Важно, что индексы в массивах и коллекциях Java начинаются с 0, а не с 1. В массиве или коллекции, состоящей из 10 элементов, индекс первого элемента 0, а последнего 9.

Рисунок 3 “Массивы и коллекции в Java”
Наследование и полиморфизм
Объектно-ориентированное программирование (ООП) - подход к созданию программ, основанный на использовании классов и объектов, взаимодействующих между собой (Рисунок 4).
Класс описывает устройство и поведение объектов. Устройство описывается через набор характеристик, а поведение – через набор доступных для объектов операций (методов). Классы можно создавать на основе уже имеющихся, добавляя или переопределяя свойства и методы.
Наследование является неотъемлемой частью Java. При использовании наследования принимается во внимание, что новый класс, наследующий свойства базового (родительского) класса имеет все те свойства, которым обладает родитель. В коде используется операнд extends, после которого указывается имя базового класса. Тем самым открывается доступ ко всем полям и методам базового класса.
Используя наследование, можно создать общий "java class", который определяет характеристики, общие для набора связанных элементов. Затем можно наследоваться от него и создать дополнительные классы, для которых определить дополнительные уникальные для них характеристики.
Главный наследуемый класс в Java называют суперклассом super. Наследующий класс называют подклассом. Таким образом подкласс - это специализированная версия суперкласса, которая наследует все свойства суперкласса и добавляет свои собственные уникальные элементы.
Полиморфизм является одним из фундаментальных понятий в объектно-ориентированном программировании наряду с наследованием и инкапсуляцией. Слово полиморфизм греческого происхождения и означает "имеющий много форм". Чтобы понять, что означает полиморфизм

применительно к объектно-ориентированному программированию, рассмотрим пример создания векторного графического редактора, в котором необходимо использовать ряд классов в виде набора графических примитивов - Square, Line, Circle, Triangle, и т.д. У каждого из этих классов необходимо определить метод draw для отображения соответствующего примитива на экране.

Рисунок 4 “Схема наследования и полиморфизма”

Особенности ООП в Java

Рассмотрим некоторые особенности объектно-ориентированного программирования (ООП) (Рисунок 5) на языке Java:

• Классы и объекты: Java поддерживает определение классов, которые служат как шаблоны для создания объектов. Объекты представляют экземпляры классов.

• Инкапсуляция: ООП в Java позволяет скрыть детали реализации и обеспечивает доступ к данным через методы (геттеры и сеттеры), что повышает безопасность и уровень абстракции.

• Наследование и полиморфизм: В предыдущем пункте мы уже подробно рассматривали данные особенности. Java поддерживает наследование, позволяя классам наследовать свойства и методы других классов. Это способствует повторному использованию кода и иерархии классов. А полиморфизм означает, что разные объекты могут иметь одинаковые методы, но действовать по-разному. Это упрощает обработку различных типов данных.

• Абстракция: Java позволяет создавать абстрактные классы и интерфейсы, которые определяют общие методы, но не предоставляют их реализацию. Это помогает создавать структуры с общими чертами.

• Пакеты: Классы и интерфейсы в Java организуются в пакеты, что упрощает управление большими проектами и предотвращает конфликты имен.

• Обработка исключений: Java предоставляет механизм обработки исключений, который позволяет более эффективно управлять ошибками в программе.

• Многопоточность: Язык Java обеспечивает поддержку многопоточности, позволяя одновременно выполнять разные части программы.

• Стандартная библиотека: Java имеет обширную стандартную библиотеку, предоставляющую множество классов и методов для разработчиков.

Приведенные особенности делают Java мощным языком программирования для разработки сложных приложений с использованием принципов ООП.

Рисунок 5 “Особенности ООП в Java”

Исключения, отладка, тестирование и логирование

Исключения в Java представляют собой события, которые могут возникнуть во время выполнения программы и прервать ее нормальное выполнение. Исключения позволяют обработать ошибки и исключительные ситуации (Рисунок 6). Для обработки исключений используются конструкции try, catch и finally.

Рисунок 6 “Исключения в Java”

Отладка - это процесс выявления и устранения ошибок в коде. В Java для отладки можно использовать интегрированные среды разработки (IDE), такие как Eclipse или IntelliJ IDEA. Они предоставляют средства для установки точек останова, шаг-за-шагом выполнения кода и анализа значений переменных (Рисунок 7).

Рисунок 7 “Пример построчной отладки в Java”

Тестирование - важная часть разработки в Java. Оно позволяет проверить, что код работает правильно. В Java используются различные фреймворки для тестирования, такие как JUnit. Тесты помогают выявить ошибки и убедиться в корректности работы программы (Рисунок 8).

Рисунок 8 “Структура тестирования в Java”

Логирование - процесс записи информации о работе приложения. В Java для логирования часто используется библиотека Log4j или SLF4J. Логи помогают отслеживать работу приложения, выявлять проблемы и анализировать его поведение (Рисунок 9).

1. «Java. Полное руководство» / Герберт Шилдт, Изд. «Диалектика-Вильямс», 2018 год, 1488 с.
2. «Язык программирования C++. Лекции и упражнения» / Стивен Прата, Изд. ООО "И.Д. Вильямс", 2012 год,

1245 с.
3. «Программирование на C# для начинающих» / Алексей Васильев, Изд. ЭКСМО, 2022 год, 592 с.
4. «Разработка веб-приложений с помощью PHP и MySQL» / Веллинг, Томсон, Изд. DiaSoft, 2002 год, 848 с.
5. Java-университет [Электронный ресурс] URL: https://javarush.com/
6. Java-краткое руководство для начинающих [Электронный ресурс] URL: https://tproger.ru/translations/java-intro-for-beginners
7. Принципы ООП на примере языка программирования Java [Электронный ресурс] URL: https://topjava.ru/blog/oops-concepts-in-java
8. Язык программирования Java [Электронный ресурс] URL: https://metanit.com/java/tutorial/1.1.php
9. Основы C++ [Электронный ресурс] URL: https://academy.yandex.ru/handbook/cpp
10. Введение в C++ [Электронный ресурс] URL: https://metanit.com/cpp/tutorial/1.1.php
11. Язык C++ [Электронный ресурс] URL: https://prog-cpp.ru/cpp/
12. Полное руководство по языку программирования C# 11 и платформе .NET 7 [Электронный ресурс] URL: https://metanit.com/sharp/tutorial/
13. Язык программирования C# [Электронный ресурс] URL: https://timeweb.com/ru/community/articles/chto-takoe-csharp
14. Учебник по C# от А до Я [Электронный ресурс] URL: https://csharp.webdelphi.ru/uchebnik-po-c/
15. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ C# В ДЕТАЛЯХ [Электронный ресурс] URL: https://itvdn.com/ru/blog/article/oop-csharp-basic
16. Справочник языка. Руководство по PHP [Электронный ресурс] URL: https://php.ru/manual/
17. Язык программирования PHP [Электронный ресурс] URL: https://skysmart.ru/articles/programming/yazyk-programmirovaniya-php
18. Руководство по PHP [Электронный ресурс] URL: https://metanit.com/php/tutorial/
19. C ++ - Классы и объекты [Электронный ресурс] URL: https://unetway.com/tutorial/c-klassy-i-obekty