

CHƯƠNG TRÌNH ĐÀO TẠO

Nhân viên: TạoKV

Thời gian: từ 28/07/2014

Người lập: Phạm Công Phan

Chương trình đào tạo (28/07/2014-28/10/2014)

Pre knowledge: OOP, JAVA.

Day(s)	Chủ đề	Nội dung	Người hướng dẫn, hỗ trợ	Đánh giá kết quả
Tổng: 5 weeks. Nộp báo cáo và bài tập	Lập trình hướng đối tượng	Hiểu các khái niệm công nghệ hướng đối tượng (Mỗi khái niệm đưa ra đều phải đưa ra ví dụ minh họa cho khái niệm đó) 1. Lập trình hướng đối tượng:	học ngoài giờ	Làm báo cáo chi tiết các kiến thức thu

<p>vào 05/09/2014</p>		<ul style="list-style-type: none"> - Các khái niệm?. - Advantages of Object-oriented Methodology? <p>2. Trừu tượng và đóng gói dữ liệu (Data abstraction and encapsulation):</p> <ul style="list-style-type: none"> - Khái niệm abstraction & encapsulation?, tại sao cần trừu tượng và đóng gói dữ liệu. Ví dụ minh họa? - Đóng gói dữ liệu trong java: Các mức đóng gói trong java? - Cách khai báo lớp, các thuộc tính, các phương thức, constructor & destructor (default constructor, copy constructor...), <p>3. Thừa kế (Inheritance):</p> <ul style="list-style-type: none"> - Khái niệm quan hệ thừa kế?, các thuật ngữ trong quan hệ thừa kế, các mức thừa kế (public, protected, private). - Multiple Inheritance? <p>4. Polymorphism:</p> <ul style="list-style-type: none"> - Đa hình là gì? - Đa hình hàm – function polymorphism? - Đa hình và hướng đối tượng? - So sánh overloading và overriding? - Method overriding? - Abstract class, interface? <p>5. Tìm hiểu 2 tiêu chí đánh giá chất lượng thiết kế module: module cohesion & coupling.</p> <ul style="list-style-type: none"> - Cohesion: Độ kết dính của một module: ám chỉ mức độ liên quan về chức năng giữa các nhiệm vụ của module. 	<p>hoạch được, mỗi phần đều có ví dụ cài đặt đơn giản để minh họa.</p>
---------------------------	--	--	--

		<ul style="list-style-type: none"> - Coupling: Độ ghép cặp giữa các module, chỉ mức độ phụ thuộc lẫn nhau của các module trong 1 hệ thống. <p>Các mục mở rộng tìm hiểu thêm (không bắt buộc).</p> <p>1. Exception handling.</p> <ul style="list-style-type: none"> - throw ngoại lệ - Cơ chế try-catch trong java. - Chuyển tiếp ngoại lệ. - Khai báo ngoại lệ. <p>2. Generic programming (Lập trình tổng quát):</p> <ul style="list-style-type: none"> - Template?, khuôn mẫu trong java, khuôn mẫu hàm (template function), khuôn mẫu lớp (template class), các tham số template. 		
	Bài tập vận dụng:	Bài tập 1, 2 phần OOP	Học ngoài giờ	Bao gồm source và chỉ dẫn biên dịch, hướng dẫn sử dụng. Trong quá trình làm bài tập cần bổ sung báo cáo thu hoạch phân lý thuyết

1 week	Tìm hiểu về mẫu Design Pattern: State	<div data-bbox="591 213 1630 335"> <ul style="list-style-type: none"> - Tìm hiểu mẫu thiết kế State - Implements class diagram sau và viết chương trình test để test hết các tính năng: </div> <div data-bbox="591 421 1617 925"> <pre> classDiagram class TCPClientConnection { - reconnectRetry: int - state: TCPClientConnectionState + changeState(TCPClientConnectionState): void + checkConnection(): int + connect(String): String + disconnect(): int + send(String): int } class TCPClientConnectionState { <<interface>> + checkConnection(): int + connect(String): int + disconnect(): int + send(String): int } class ConnectedState { + checkConnection(): int + connect(String): int + ConnectedState(): void + disconnect(): int + finalize(): void + send(String): int } class ConnectingState { + checkConnection(): int + connect(String): int + ConnectingState(): void + disconnect(): int + finalize(): void + send(String): int } class DisconnectedState { + checkConnection(): int + connect(String): int + disconnectedState(): void + disconnect(): int + finalize(): void + send(String): int } TCPClientConnectionState < -- ConnectedState TCPClientConnectionState < -- ConnectingState TCPClientConnectionState < -- DisconnectedState TCPClientConnection o-- TCPClientConnectionState </pre> </div> <div data-bbox="591 970 1617 1367"> <p>*Mô tả: TCPClientConnection là đối tượng trừu tượng hóa của 1 kết nối TCP phía client lên server. Connection luôn có 3 trạng thái, Disconnected khi Connection vừa khởi tạo hoặc bị mất kết nối với server. Connecting là trạng thái mà Connection đang (try) kết nối tới server, Connected là trạng thái mà Connection đã tạo kết nối thành công đến server.</p> <ul style="list-style-type: none"> - Hàm TCPClientConnection::checkConnection() là hàm được gọi định kỳ bởi 1 timer để kiểm tra trạng thái kết nối giữa client và server: + Nếu trạng thái hiện tại là connected thì sẽ gửi bản tin heartbeat lên server + Nếu trạng thái hiện tại là connecting thì tăng biến đếm số lần retry </div>
--------	---------------------------------------	---

		<p>reconnect tới server, nếu số lần retry reconnect > 5 thì sẽ chuyển trạng thái Connection về Disconnected. (nếu đang ở trạng thái connecting mà nhận được event Connection Connected do tầng network notify lên thì chuyển trạng thái Connection sang Connected).</p> <p>+ Nếu trạng thái hiện tại là disconnected thì sẽ thực hiện reconnect (connect) tới server và chuyển trạng thái sang connecting.</p> <ul style="list-style-type: none"> - Hàm send(String message) chỉ được phép khi Connection ở trạng thái Connected, các trường hợp khác sẽ báo lỗi (exception). - Hàm close() chỉ hoạt động khi Connection ở trạng thái Connected hoặc Connecting. 		
--	--	---	--	--

Java Core

Day(s)	Chủ đề	Nội dung	Hướng dẫn, hỗ trợ	Đánh giá kết quả
Tổng: weeks	Lập trình java nâng cao	<p>Yêu cầu:</p> <ul style="list-style-type: none"> - Mỗi phần khái niệm đưa ra đều phải có ví dụ minh họa. - Trong mỗi bài tập áp dụng đều phải có mô tả ngắn về cách giải quyết bài toán, phần mã chương trình có comment đầy đủ + dễ hiểu. - Làm bài tập trong file: Java Programming Exercise.doc - Xem trước các yêu cầu bài tập. Trong quá trình học các phần được yêu cầu, cần map được các 	Học tại công ty và học ngoài giờ	Làm báo cáo chi tiết các kiến thức thu hoạch được, mỗi phần đều có ví dụ cài đặt đơn giản để minh họa.

		kỹ năng đã học thì giải quyết được vấn đề gì trong các yêu cầu đó.		
(2 weeks) 28/07/2014 – 11/08/2014	Streams and Input/Output	<ul style="list-style-type: none"> - File-Management Methods? Tìm hiểu các lớp File, <i>RandomAccessFile</i> - Tìm hiểu các lớp Input Streams và Readers, cho ví dụ? - Tìm hiểu các lớp Output Streams và Writers, cho các ví dụ so sánh? - Object Persistence and Serialization? - Logging: Tìm hiểu và sử dụng gói Log4j do Apache cung cấp. - Tự thực hiện các bài tập minh họa - Làm bài tập 1 phần Java I/O 		
11/08/2014 – 24/08/2014	Java New I/O Package	<ul style="list-style-type: none"> - Tìm hiểu Java NIO theo tài liệu đi kèm: NIO Buffers, Channels, and Character Sets? - Có report báo riêng về phần này. - Tự thực hiện các bài tập minh họa 		
25/08/2014 – 05/09/2014	Threads and multithread basics	<ul style="list-style-type: none"> - Creating and Running a Thread? - The Thread-Control Methods? - The Thread's Life Cycle (nêu các trạng thái của một thread, giải thích?)? - Thread Priorities and Scheduling: 		

		<ul style="list-style-type: none"> + So sánh 2 phương thức Preemptive Scheduling và Time Slice Scheduling. + Thread Priority Values? + Cách tạo và sử dụng periodic tasks sử dụng lớp java.util.Timer và java.util.TimerTask ? - Làm bài tập 1. 		
06/09/2014 – 18/09/2014	Thread synchronization	<ul style="list-style-type: none"> - Conception of deadlock, starvation? - Vì sao cần có thread synchronization? - Nêu các trường hợp sử dụng từ khóa "synchronized" và ý nghĩa sử dụng trong các trường hợp đó? Cho ví dụ minh họa với mỗi trường hợp? - So sánh synchronized method vs synchronized block, class-based synchronization vs object-based synchronization? - Bài tập 2 		
19/09/2014 – 29/09/2014	Inter-thread Communications	<ul style="list-style-type: none"> - Nêu ý nghĩa của việc sử dụng các phương thức wait(), notify() và notifyAll() (đây là cơ chế đồng bộ hóa phức tạp hơn trong các giao tiếp giữa các thread trong 1 chương trình), cho 1 ví dụ minh họa sử dụng các phương thức này? - Using join() to Wait for a Thread to Die? 		

		- Bài tập 3		
30/09/2014 – 05/10/2014	Threadpool & RMI	<p>Tìm hiểu các mô hình Thread Pool (Pooling Thread model).</p> <p>Tìm hiểu lập trình RMI trong java</p>		
	Java SE và Webservices	<p>Tìm hiểu các mô hình webservises: SOAP, RESTful:</p> <ul style="list-style-type: none"> - Phân biệt sự khác nhau và các tình huống vận dụng tốt nhất - So sánh về hiệu suất giữa 2 mô hình này (với cách lập trình thông thường) <p>Code chương trình test WS sử dụng SOAP và RESTful (server).</p> <p>Code chương trình client và gọi tới (Webservice) server bên trên.</p>		
06/10/2014 – 28/10/2014	Làm bài tập tổng hợp	Project 4		Làm bài tập tổng hợp các kiến thức đã đào tạo + tự đào tạo.

				Cần xem trước yêu cầu của bài tập từ trước
--	--	--	--	--

1.1 Một số yêu cầu.

Nắm các kiến thức về các nội dung đề ra trong plan.

Làm các bài tập sau mỗi phần (theo plan hoặc tự đề ra theo kiến thức học tập)

Có Báo cáo kết quả thu hoạch sau mỗi phần. Cần chi tiết các phần trọng tâm. Các báo cáo được viết tiếp vào các phần trong tài liệu mẫu báo cáo gửi kèm

Lập và kiểm soát kế hoạch theo quy định của công ty: Lập kế hoạch tuần vào thứ 2 và báo cáo vào thứ 6

Trao đổi với người hướng dẫn hoặc các thành viên khác khi gặp khó khăn.

Một số lưu ý khi lập trình: Theo file hướng dẫn gửi kèm.