```
package slashuscraper.analyze;

import java.time.LocalDate;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.concurrent.Callable;

import slashuscraper.object.Comment;
import slashuscraper.object.User;
import slashuscraper.object.Word;

public class AnalyzeUser implements Callable<User> {

	private User user = null;

	public AnalyzeUser(User user) {
		this.user = user;
	}

	@Override
	public User call() throws Exception {

		// Join date, based on first comment/post
		LocalDate joinDate = LocalDate.now();

		// Temporary user comment attributes
		Comment comment = null;
		LocalDate date = null;

		// Process all user comments and posts
		while((comment = user.popComment()) != null) {

			// System.out.println("[STATUS] AnalyzeUser processsing comment");

			// Get date posted
			date = comment.getDatePosted();
			// Replace join date if earlier
			if(date.isBefore(joinDate)) {
				joinDate = date;
			}

			// System.out.println("[STATUS] Date: " + date.getDayOfWeek().getValue());

			// Increment post per day of the week
			user.addToPostRate(date.getDayOfWeek().getValue());

			// System.out.println("[STATUS] Added date");

			// Check if content exists
			if(comment.getContent().length() > 0) {
				// Create word rate iterator
				Iterator<Entry<String, Word>> itr = comment.getWordFrequency()
							.entrySet().iterator();
				Map.Entry<String, Word> entry = null;
				// Add words and counts to word rate
				while(itr.hasNext()) {
					// Get next entry
					entry = itr.next();
					// Add values to existing word bank
					user.addWordUsed(entry.getValue().getWord(), entry.getValue().getCount());
				}
			}

			// Add karma elemnts to user average (upvotes - downvotes)
			user.addKarmaToAverage(comment.getUpvotes() - comment.getDownvotes());

			// Add comment sub to list of subs visited
			user.addVisitedSub(comment.getSubreddit());
		}

		// Set join date
		user.setJoinDate(joinDate);

		// Return processed user
		return this.user;
	}
}
```