Java Full Stack Developer Q&A - Part 6: Behavioral Questions (Q151–180)

## Q151: Tell me about a time you handled a critical production issue.

In my role at Amazon Robotics, a production microservice handling delivery data started throwing 500 errors due to Kafka message overload. I quickly analyzed logs using Kibana, identified a memory leak in message consumption, and deployed a hotfix by optimizing the listener. We introduced batch processing and backpressure handling, restoring stability within 2 hours.

## Q152: How do you manage conflicts within your development team?

I always encourage open communication. At Biogen, there was a disagreement between frontend and backend teams on API design. I facilitated a joint meeting, collected each side's constraints, and proposed a contract-first API design using Swagger, which satisfied both parties.

## Q153: Describe a situation where you had to quickly learn a new technology.

While working at Dell, we needed to migrate a legacy monolith to Kubernetes. I had limited exposure to Kubernetes, so I dedicated evenings to Coursera training, hands-on labs, and internal sandbox testing. Within 2 weeks, I led containerization and deployed the app using Helm charts.

## Q154: How do you handle tight deadlines without compromising quality?

I use Agile techniques like breaking work into prioritized user stories. At Biogen, we had a 2-week deadline for a compliance audit dashboard. I focused on delivering the core MVP first with test coverage and automated CI/CD. This allowed us to demo early and incorporate feedback quickly.

## Q155: What do you do if you disagree with a product owner’s requirements?

I validate their reasoning and back my concerns with data or feasibility analysis. In one instance, a feature was expected to query millions of records in real time. I provided load testing metrics and proposed a caching and pagination strategy, which was accepted.

## Q156: Tell me about a time you improved the performance of an application.

At Dell, a reporting module was taking over 60 seconds to load. I used APM tools and DB query analysis to identify bottlenecks. I optimized SQL queries, introduced Redis caching, and parallelized heavy computations, reducing load time to under 5 seconds.

## Q157: How do you stay current with evolving technologies?

I dedicate weekly time for learning through online courses (Pluralsight, Udemy), follow GitHub open-source projects, attend webinars, and contribute to tech forums. This habit helped me quickly adopt Java 17 and Angular 15 in recent projects.

## Q158: How do you ensure code quality in your team?

I enforce coding standards through SonarQube, peer code reviews, and automated tests. At Amazon Robotics, we maintained 90%+ unit test coverage and used Git hooks to enforce style checks and coverage before merge.

## Q159: Describe a time you led a cross-functional team.

During the robotic tracking initiative at Amazon, I coordinated efforts between DevOps, QA, frontend, and analytics teams. I created a shared Jira dashboard, held daily syncs, and used Confluence to track decisions, which ensured alignment and successful delivery.

## Q160: What do you do when a project’s requirements change suddenly?

I reassess the backlog and reprioritize with the product owner. For example, mid-sprint at Biogen, a compliance update was mandated. I negotiated to move less critical features to the next sprint and allocated time for the update without overburdening the team.

## Q161: How do you deal with repetitive tasks?

I automate them. I’ve written bash scripts and Jenkins jobs to automate test runs, deployment, and log archiving—saving hours each week.

## Q162: How do you mentor junior developers?

I set weekly one-on-ones, pair programming sessions, and code walkthroughs. I also assign small feature ownership with my supervision to grow their confidence.

## Q163: Tell me about a mistake you made and what you learned.

I once deployed a database patch without dry-running it in staging. It caused a failure in a dependent service. Since then, I’ve implemented mandatory staging dry runs and rollback scripts for all DB changes.

## Q164: How do you prioritize features?

I consider business impact, complexity, and risk. Using MoSCoW technique and consulting with stakeholders, we prioritize must-have features first, then nice-to-haves.

## Q165: How do you manage stress in high-pressure environments?

I stay calm, focus on one task at a time, and rely on my team. I also practice time blocking and avoid unnecessary multitasking.

## Q166: How do you handle unclear requirements?

I schedule clarification meetings, create wireframes or prototypes to validate understanding, and iterate based on stakeholder feedback.

## Q167: Describe a time you received critical feedback.

A manager once pointed out I was over-engineering a module. I acknowledged it, simplified the design, and adopted the ‘You Aren’t Gonna Need It’ principle going forward.

## Q168: What motivates you in your work?

Solving complex problems, building scalable systems, and mentoring others. I enjoy seeing my code drive business outcomes.

## Q169: How do you handle multiple projects at once?

I use project management tools like Jira and Trello, time-block my calendar, and set clear boundaries for context switching.

## Q170: What’s your leadership style?

Servant leadership. I believe in empowering my team, removing blockers, and fostering a culture of ownership and trust.