AI in IntelliJ

Slides on GitHub

Part One

Smarter (usually) auto-complete

```
alignedAccountsUpdated {
  policyNumber: number,
  processDate: date,
  effectiveDate: date,
  adjustmentTypeCode: string
  paymentDate: date,
  cashReceiptNumber: string,
  cashRecipiTDate: date,
  paidAmount: number,
  receiptType: string,
  receiptNumber: string
}[ ]
  policyNumber: 30012 /* TBC in consumer
  processDate: rss.Date(),
  effectiveDate: null,
  adjustmentTypeCode: '01-01/MTM',
  paymentDate: rss.Date(),
  cashReceiptNumber: '1234567890',
  cashRecipiTDate: rss.Date(),
  paidAmount: 100,
  receiptType: '1234',
  receiptNumber: '1234567890'
};
```

```
alignedAccountsUpdated {
  policyNumber: number,
  processDate: date,
  effectiveDate: date,
  adjustmentTypeCode: string
  paymentDate: date,
  cashReceiptNumber: string,
  cashRecipiTDate: date,
  paidAmount: number,
  receiptType: string,
  receiptNumber: string
}[ ]
  policyNumber: 30012 /* TBC in consumer
  processDate: rss.Date(),
  effectiveDate: null,
  adjustmentTypeCode: '01-01/MTM',
  paymentDate: rss.Date(),
  cashReceiptNumber: '1234567890',
  cashRecipiTDate: rss.Date(),
  paidAmount: 100,
  receiptType: '1234',
  receiptNumber: '1234567890'
};
```

policy-number-util.ts | policy-number-normalization-result.ts

```
export class PolicyNumberNormalizationResult {

  inputPolicyNumberString: string;
  normalizedPolicyNumberString: string;

}
```

policy-number.proto | policy-number-normalization-result.ts | policy-service.ts | application.properties

```
export class PolicyNumberNormalizationResult {

  inputPolicyNumberString: string;
  normalizedPolicyNumberString: string;
  isPolicyNumberValid: boolean;
  errorMessage: string;

  constructor(
    inputPolicyNumberString: string,
    normalizedPolicyNumberString: string,
    isPolicyNumberValid: boolean,
    errorMessage: string) {
    this.inputPolicyNumberString = inputPolicyNumberString;
    this.normalizedPolicyNumberString = normalizedPolicyNumberString;
    this.isPolicyNumberValid = isPolicyNumberValid;
    this.errorMessage = errorMessage;
    return new;
  }

}
```

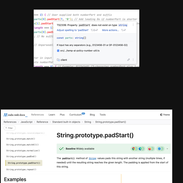
```
{ one }
  return new PolicyNumberNormalizationResult(inputPolicyNumber, "Character count too low to make a valid policy", false);
}
```

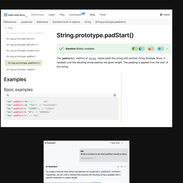
```
{ one }
  return new PolicyNumberNormalizationResult(inputPolicyNumber, "Character count too 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
```

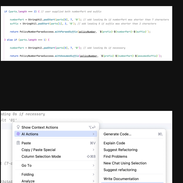
Part Two

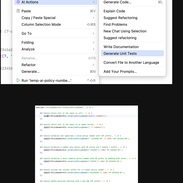
Asking for help & "Fix it for me"

const validateEmail = (email: string): boolean =>
export interface ValidationResult {
isValid: boolean;
errorMessage?: string;
}
export const validateEmail = (email: string): ValidationResult =>
validateEmail(email) ? { isValid: true } : { isValid: false, errorMessage: 'Invalid email format' };









Going from TypeScript back to Groovy was less successful

Part Four

Misc

Explain this code

Good enough to email to another programmer.

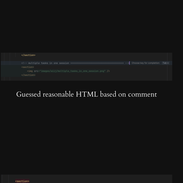
Only change required was
removing some obvious points.

Explain error

Asked about Grails upgrade and queue integration.
Answers were too generic and not helpful
("here's how to setup Grails" or
"here's what to put in config for queue"),
suggesting stuff I had already done or
not pertinent or
for older/other versions of libraries.

Part Five

Help with these slides



Guessed reasonable HTML based on comment

Inserts random broken link

ROBOL (Random COBOL?)

CSS help w/o StackOverflow

Weird English autocomplete

Part Six

Part Six

Notes

Originally graded these

A = No changes from me
B = Mostly right, but needs tweaks
F = Wasted time
Probably a B on average

Privacy concerns

Clients okay with this?
Code goes to partners, but
agreements in place not to train or share

Options

Built-in local full-line completion
JetBrains AI Assistant plugin
Third-party plugins
Soon: Junie (agentic)

Languages

TypeScript support is good
Groovy is 💩
Java would be good

Refactorings

Generally not helpful
I'm already good at refactoring
IntelliJ is already good at refactoring
could != should
Technically correct but not necessary

Article on MartinFowler.com

https://martinfowler.com/articles/exploring-gen-ai.html
Lack of reuse
Verbose and redundant tests

Stumbling block

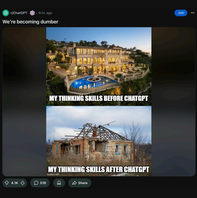
Open-ended nature of requests
Mitigated by discrete commands in context menu
and other UI affordances
Chat interface requires different way of thinking

Versus Google, StackOverflow

Quicker than Google/StackOverflow *for some things*,
but oftentimes not b/c of how wordy it is,
depends on how focused/targeted your search is.

(Ironically, by Google making itself worse for searches,
AI—both here and in Google—have become more
useful).

AI versus thinking



AI versus taking a walk

AI vs An extra minute in the shower
👎 We should be thinking through these problems
👍 Saves an extra minute or two of thought
👍 Saves time typing (more time for thinking)
Second set of eyes (pair programming) 👀