

本章概要

概述:

我们将使用Fitten Code辅助编程,让编程变得更简单



- 作为现代编程人员,我们总是追求更快、更高效的工作方式。使用AI 编程助手如同拥有一个可靠、智能的合作伙伴,它能够与你紧密合作,提供实时的建议和解决方案。无论是快速修复错误、提升代码质量,或者查找关键文档和资源,AI编程助手都能让你事半功倍。让我们携手AI编程助手,释放创造力,加速项目进程,共同迈向编程的新高度!
- Fitten Code是由非十大模型驱动的AI编程助手,它可以自动生成代码,提升开发效率,帮我们调试Bug,节省我们的时间。还可以对话聊天,解决我们编程碰到的问题。免费且支持80多种语言:Python、C++、Javascript、Typescript、Java等,目前支持以下4种编辑器与开发环境:

VS Code

JetBrains IDE 系列

Visual Studio

Vim

Va

代码自动补全

• Fitten Code 能够自动为开发者的代码补充缺失的部分,节我们宝贵的 开发时间。

注释生成代码

• Fitten Code 基于AI大模型对代码进行语义级翻译,支持多种编程语言 互译。

自动添加注释

• Fitten Code 能够根据代码自动生成相关注释,为我们的代码提供清晰 易懂的解释和文档。

智能bug查找,解释代码,自动生成单元测试的功能,根据代码自动产生相应的测试用例等。

更多功能

15.2 安装

 如果我们已经安装JetBrains系列产品且版本大于等于2021.1,请直接 跳过此步骤。否则请前往官网下载安装最新JetBrains系列产品。如果 我们的版本低于2021.1,插件市场将无法搜索到Fitten Code插件,请 升级到最新版本。



15.2 安装

•1 打开IDE, 点击左上方 "File" 接着点击 "Settings"

۵	<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	<u>N</u> avigate	<u>C</u> ode	<u>R</u> efactor	<u>B</u> uild	R <u>u</u> n	Tools
		<u>N</u> ew				>			
8.		<u>O</u> pen. <u>R</u> ecen	 t Projec	cts		>			
00		– Close	Project						
		Remo	te Deve	lopment					
[ŝ	Se <u>t</u> tin	gs		Ctrl+	Alt+S			
	G	Reloa	d CMak	e Project					
		File Pr	opertie	s		>			
		Local	History			>			

15.2 安装

• 3 安装完毕后点击左侧Fitten Code小图标进行登录后即可使用

⑧ 用户名登录	. 手机号登录	□ 邮箱登录
用户名		
密码		
忘记密码		注册
	登录	
	──── 其他登录/注册方式 ──	

15.2 安装

• 2 点击"Plugins"然后选择"Marketplace",随后在搜索框中输入"Fitten Code", 再点击"Install"

Settings	③ 输入Fitten Code	2 Marketolace
	Plugins	
> Appearance & Behavior	Q- Fitten Code	×
> Editor	Search Results (1)	Sort By: Relevance 👻
Plugins 1	Fitten Code	Install
> Version Control	Fitten Tech.	/
> Build, Execution, Deployment		4
> Languages & Frameworks		

15.2 安装

• 4 如果我们出现了"请安装JCEF"的错误,将会使得Fitten Code无法正常工作,请按照以下步骤进行操作:

(a). 在菜单栏, 点击 Help | Find Action

(b). 搜索并运行"Choose Boot Java Runtime for the IDE";

(c). 选择带有 JCEF 的 Java 运行环境(这一步需要联网,请确保我们的网络和代理服务器畅通); Choose Boot Runtime for the IDE

(d). 等待IDEA更新并重启。

A Chang it unles	A Changing the runtime may cause unexpected problems. Do not change it unless you were specifically asked to do so by JetBrains support.							
Current:	17.0.4.1+7-469.62-jcef JetBrains Runtime Bundled							
New:	Select runtime							
Location:	The selected boot runtime will be downloaded if needed and set for the current IDF							
Use De	efault Runtime OK							

15.2 安装

- 5 如果我们出现了"Too many restarts of GPU-process"的错误,并且Fitten Code侧边栏无 法显示,请按照以下步骤进行操作:
- (a). 在菜单栏, 点击 Help | Find Action;
- (b). 搜索并运行"Registry...";
- (c). 找到并关闭 "ide.browser.jcef.sandbox.enable" 的勾选;
- (d). 确定并重启IntelliJ IDEA。

E Registry	×
▲ Changing these values may cause unwanted behavior of IntelliJ IDEA. Do not cha Ø 5	inge these unless you have been asked to.
Кеу	Value
ide.browser.jcef.osr.wheelRotation.factor	10
ide.browser.jcef.preinit \Lambda	
ide.browser.jcef.sandbox.enable 🔥	
ide.browser.jcef.svg-viewer.debug	
ide.browser.jcef.svg-viewer.enabled	
ide.browser.jcef.testMode.enabled	
ide.built.in.web.server.activatable	
Enable sandbox in JCEF Requires an IDE restart	
	Restore Defaults Close

15.3 智能补全

• 打开代码文件,输入一段代码, Fitten Code就会为我们自动补全代码

	import torch	
	<pre>class ResNet(torch.nn.Module):</pre>	
4	<pre>definit(self, num_classes=10):</pre>	
	super(ResNet, self)init()	
	<pre>self.conv1 = torch.nn.Conv2d(3, 64, kernel_size=7, stride=</pre>	2, padding=3, bias=False)
	self.bn1 = torch.nn.BatchNorm2d(64)	
	self.relu = torch.nn.ReLU(inplace=True)	
	<pre>self.maxpool = torch.nn.MaxPool2d(kernel_size=3, stride=2,</pre>	padding=1)
	self.layer1 = selfmake_layer(64, 64, 3)	
	<pre>self.layer2 = selfmake_layer(64, 128, 4, stride=2)</pre>	
	self.layer3 = selfmake_layer(128, 256, 6, stride=2)	



• 用户可通过点击左上角工具栏中的"Fitten Code - 开始对话"或者使用快捷键 "Ctrl+Alt+C"打开对话窗口进行对话

	9 D
뎞 New Chat	
Reply	
	e ū

15.5 生成代码

• 生成代码



15.5 生成代码

• 生成代码

	🗮 📫 intellij-platform-plugi	n-template-main ~ 版本控制 ~		
			G % D	
	🎢 Generate Code			
80	帮我生成 python 快速排序的代码	输入指令后回车即可生成		
	<pre>def quicksort(arr): if len(arr) <= 1: return arr pivot = arr[len(arr) // left = [x for x in arr middle = [x for x in arr right = [x for x in arr return quicksort(left) arr = [3, 6, 8, 10, 1, 2, print(quicksort(arr))</pre>	<pre>// 2] r if x < pivot] arr if x == pivot] rr if x > pivot] 0 + middle + quicksort(right) 1]</pre>	Сору	点击复制
	Reply			
			•	

15.6 代码翻译

• 解释代码(选中代码段后右键,后续操作同理)

	■ 🛤 intellij-platform-plugin-template-main ~ 総合控制 ~				
		💮 МуТ	foolWindowFactory.kt ×		
80	Come and experience efficient code completion! Open the code file you are writing and enter any code to use the auto completion function. Press bab to accept all completion suggestions. Press Cirl+ I to accept a line of completion suggestion. Press Cirl+ I to accept a single word completion suggestion. You can also click on the "chat icon•" above to experience practical small features. For example, click on the chat icon above to start chatting with the Al code assistant.		<pre>package org.jetbrains.plugins.template.toolWindow > import</pre>		
	If there are any Fitten Code related issues or suggestions, Welcome to click here for feedback		<pre>overrids fun createToolkindowCentent(project: Project, toolWindow: ToolWindow) { val avToolWindow = MyToolWindow(toolWindow) val content = ContentFactory.getInstance().createContent(eyToolWindow.getContent toolWindow.contentEnanager.addContent(Content) overrids fun shouldBeAvailable(project: Project) = true class MyToolWindow(toolWindow: ToolWindow) { private val service = toolWindow.project.service<ryprojectservice>() fun getContent() = JBPanel<jbpanel<*>().apply { val tabel = JBLabel(RyBundle.nessage("randomLabol*, "?")) add(tabel) add(tabel) add(tabel) addActionListemer { label.text = MyBundle.nessage("randomLabol*, service.getRandomNumber }) } } } </jbpanel<*></ryprojectservice></pre>	 Fitten Code ○ 知兆上下文時作 ○ 知兆上下文時作 ○ 気約(C) ② 気約(C) ③ 私約(C) ③ 私約(C) ③ 私約(C) ③ 私約(C) ○ 抗約(Ab145株 月之廿祝式(M) ○ 衣火井中白我 白我司法(J) 特別 若査 七州 Google 技術(S) 年秋(B) 主城 打开丁 Local History (学 与羽和和比較(B) 	Fitten Code - Explain Code Fitten Code - Start Chat Fitten Code - Find Bugs Fitten Code - Generate Code Fitten Code - Generate Unit Test Fitten Code - Document Code Fitten Code - Edit Code

15.6 代码翻译

• 解释代码(选中代码段后右键,后续操作同理)

🛔 🗮 🚺 intelių-platform-plugin-template-main 🐃 MLA-2014 🛸

<pre></pre>				
Concentration Concent	1		- MyTo	oolWindowFactory.kt ×
<pre>Bigmands The product outs rupped is non-marked body applying all contains as work of the four state is contained to a body applying all contains as work of the four state is contained to a body applying all contains as the contained applying applying</pre>		G × c	Ð	package org.jetbrains.plugins.template.tool#indow
<pre>prove the function for the product of the bin active field and product of the bin active field field product field fiel</pre>	IB Explain Co	de		
<pre>Luts my for landow fac node may by spin: </pre>	The provided averride fun	code snippet is from a Kotlin file, likely within an Intelli# IDEA plugin, and it contains an ction to create the content of a tool window within the IDE.		
<pre>int f a creater balandouckerter (regist; regist; regist;</pre>	Let's break do	wn the code step by step:		class MyTcolWindowFactory : ToolWindowFactory {
<pre>improvements of the improvements of the improvement of the improvements of the improvements of the im</pre>	1. override This line do Toolkindo with the p	<pre>fun createToolWindowContent(project: Project, toolWindow: ToolWindow) { edares a function createToolWindowContent which overrides a function defined in the swFactory interface. When a tool window is created within the IDE, this function is called oject and tool window as parameters. Subjective - W-ToolWindowContent(interve)</pre>		<pre>init { thisLogger().warn("Gon't forget to remove all non-needed sample code files with their corresponding registration entries in "plugin.xml",") }</pre>
<pre>statusts is not is not its is a partial is a partial</pre>	Here, a ner	w instance of the MyToolWindow class is created, passing the toolWindow from the function		override fun createToolWindowContent(project: Project, toolWindow; ToolWindow) {
<pre>totac: isles</pre>	pårameter 3. vel contr	s to its constructor. wrt =		<pre>vel myToolWindow = MyToolWindow(toolWindow) vel content = ContentFactory.getInstance().createContent(nyToolWindow.getContent(), null, false)</pre>
<pre>Nin Successing the content for the tool window. Busing the ContentFactory to crast a content intense using the content for the Ny Toolkindow. Content many, add content in added by the contentFactory to crast a content intense using a content in the Hy Toolkindow. Content many, the medy content many, add content in added by the contentFactory add content to the ool window by creater Toolkindow. Content many, the work is content for the tool window by creater Toolkindow. The many the content for the tool window by creater Toolkindow. The many the medy content many, the My Toolkindow dask (bindow by content the bool window is content many). Townide/Himmat(ta) number to the tool window, including UI composers and event handles. many the first the tool window, including UI composers and event handles. many the first the tool window, including UI composers and event handles. many the first the tool window, including UI composers and event handles. many the first the tool window, including UI composers and event handles. many the first the tool window, including UI composers and event handles. many the first the tool window, including UI composers and event handles. many the first the tool window, including UI composers and event handles.</pre>	false)	<pre>sctory.getInstance().createcontent(ay/dolwindow.getContent(), null,</pre>		toolWindow.contentManager.addContent(content)
<pre>A toolkindow.contentionager.addContent(content) Toolkindow.contentionager.addContent(content) Toolkindow.contentionager add the contentinanger of the toolkindow. In summary, the main function of this code snippet is to override the createToolkindowContent function and provide the content to the tool window (rotating an instance of MyToolkindow and adding its content to the book indow(southing UI components and event handles. verrideKinfHildft: Distribution OverrideKinfHildft: CdsSifty/GPU: vertify add(JBUTCONCONTENT (AddSifty/SPU: Vertify add(JBUTCONCONTENT) (AddSifty/SPU</pre>	This line or instance u	eates the content for the tool window. It uses the ContentFactory to create a content ing the content of the MyToolWindow instance created in the previous step.		anardida dua abad diadaati adiadaati. Beedaati a kalla
<pre>hs unmanage, the main function of this code snippet is to override the created oslikindow(toolWindow) conting an instance of MyToolWindow and beyoint to the tool window, including UI components and event handles.</pre> class MyToolWindow(and the code snippet is to override Window) conting an instance of MyToolWindow and beyoint to the tool window, including UI components and event handles. class MyToolWindow(and the code snippet is to override Window) conting an instance of MyToolWindow and the code snippet is to override Window) content to the tool window, including UI components and event handles. class MyToolWindow(and the code snippet is to override Window) content to the tool window, including UI components and event handles. class MyToolWindow(and the code snippet is to override Window) content to the tool window, including UI components and event handles. class MyToolWindow(and the code snippet is to override Window) content to the tool window, including UI components and event handles. class MyToolWindow(and the code snippet is to override Window) content to the tool window, including UI components and event handles. class MyToolWindow content to the tool window, including UI components and event handles. class MyToolWindow content to the tool window, including UI components and event handles. class MyToolWindow content to the tool window (toolWindow) content to the tool window) content to the tool window (toolWindow) content to the tool window (tool window) content to the	4. toolWinds Finally, the	w.contentHanager.addContent(content) newly created content is added to the contentHanager of the toolkindow.		
www.ideff/HBMHAN DBMRERIO override XBF7HF1GTX+0m55 (Bas) Qxx.08C10+CUX/D9B8X0.08H. fun getContent() = .J8Panel <j8panel<-(.3panel< td=""> GESDB9:000+ override XBF7HF1GTX+0m55 (Bas) Qxx.08C10+CUX/D9B8X0.08H. add(label) add(label) add(label) musp/mainting</j8panel<-(.3panel<>	In summary, th function and p adding its con specific content	we main function of this code snippet is to override the createToolWindowContent wovide the content for the tool window by creating an instance of MyToolWindow and tent to the tool window's content manager. The MyToolWindow class likely defines the st and behavior of the tool window, including UI components and event handlers.		class MyToolWindow(toolWindow: ToolWindow) { private val service = toolWindow.project.service <myprojectservice>()</myprojectservice>
override X#YRETCTSCHERG 01212200 override X#YRETCTSCHERG 01212200 override X#YRETCTSCHERG 01212200 override X#YRETCTSCHERG 01212200 Will debel add(label) add(label) add(label) y y y y y y y y y y y y y y </td <td></td> <td></td> <td></td> <td><pre>fun setContent() = JBPanel<=>>().apply {</pre></td>				<pre>fun setContent() = JBPanel<=>>().apply {</pre>
override 关键字用于在于关中面写 (截备) 少美或器口中已定义的函数或器性。 add(label) 在检查的示例中, override 关键字表明 createToolkindowContent 函数在实现 ToolkindowFactory add(label) 面面的作用在于附供一个面的实现, 以满足物量子关的需求或原则, 通过面与交类或器口中的函数或器性, 在助于面解代函数的基础器性, 有助于面解代函数的基础器性, 有助于面解代函数的基础器性, 有助于面解代函数的基础器性, 有助于面解代函数的基础器性, 有助于面解代函数的基础器性, 有助于面解代函数的基础器性, 有助于面解代函数的基础器性, 非正式, 非正式, 非正式, 非正式, 非正式, 非正式, 非正式, 非正式	override(9)117	開什么可能使提问		<pre>val label = JBLabel(MyBundle.message("randomLabel", "?"))</pre>
在設計的学校中, overlide 关键字表明 creatToolkindowContent 語数在实现 ToolkindowFactory 接口時接触写。这曲林再在该特定学中, creatToolkindowContent 語数件覆盖读服口中的本的实现。 面包的作用在于提供一个面的实现。以满正特定子生的需求必是可能有必是呢,通过面写交先或能口中的函数或属 性、子类可以定时代方为。使用符合特定的上下又或需求。 在 Kotlin中, 使用 overlide 关键字来面写交先或能口中已定义的函数或思维性,有助于理解代和的包括用的 实现,并且可以提高代和的可读性和时能全性。	override XS	[字用于在子类中重写 (覆盖) 父类或接口中已定义的函数或属性。		
中国的作用任于提供一个面的原则。以其能特定于关的需求规模用,通过重写交先或能口中的函数或属 性、子类可以定制化行为,使且符合特定的上下交或需求。 在 Kotlin中,使用 override 关键字来面写交先或能口中已定交的函数或属性,有助于理解化和的性格的 实现。并且可以提高化和的可读性和时能学校。	在检查的示例	中, override 天赋字表明 createToolWindowContent 函数在实现 ToolWindowFactory 这篇味着在该特定失中, createToolWindowContent 函数将覆盖该接口中很本的实现。		add(labet) add(JButton(MyBundle.message("shuffle")).apply { adddetion(istemen d
在 Kotlin中, 使用 over 140 关键字来面写父亲级思口中已定义的函数或思维,有助于理解代码的结构的 实现。并且可以提及代码的可读性和可能全性。	重写的作用在 性。子类可以	于提供一个新的实现,以满足特定子头的需求或原则。通过重写少类或组口中的函数或属 运制化行为,使具符合特定的上下文或需求。		<pre>label.text = MyBundle.message("rondomLabel", service.getRandomNumber()) }</pre>
Feedyname and a second se	在 Kotlin中,f 实现,并且可	e用 override 关键字来面写父类或短口中已定义的函数或属性,有电于理解代码的结构和 刘焜高代和短句话性和同情的性。		10 1
	Jeeply			

15.7 生成注释



9	■ Intellij-platform-plugin-template-main ~ 版本投制 ~		<i>E</i> 7 Rat
5		💮 MyToolWindowFactory.kt 🔍	
80	Come and experience efficient code completion! Open the code file you are writing and enter any code to use the auto completion function. • Press tab to accept all completion suggestions. • Press tab '= to accept all inne of completion suggestions. • Press Ctrl + = to accept a single word completion suggestion. You can also click on the 'dust icon • above to experience practical small features. For example, click on the chat icon above to start chatting with the All code assistant. If there are any Fitten Code related issues or suggestions, Welcome to click here for feedback	<pre>package org.jstbrains.plugins.template.tool%indow import imp</pre>	
		<pre>58 ctass MyTabl#indow(toot#indow; Tool#indow) { 51 52 53 54 55 55 55 55 55 55 55 55 55 55 55 55</pre>	atain Code et Chiat d Bugs herate Code herate Unit Test cument Code t Code
	服 务	② 与劳味果比较(B)	

15.7 生成注释



	intellij-platform-plugin-template-main v (xx4-j2.0) v		
3		G MyToolWindowFactory.kt ×	
	<pre>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>></pre>	<pre>>> package org.jetbrains.plugins.template.toolWindow >>> import >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>></pre>	
	<pre>// Adds a label with a random message val label = 2BLabel(MyBundle.message("randomLabel", "?")) add(label) // Adds a button to shuffle the label's test add(2Button(MyBundle.message("shuffle")).apply { // Updates the label's text with a random number from the servic addActionListener { label.text = MyBundle.message("randomLabel", service.getRand }) } } Copy</pre>	<pre>is toolWindow.contentManager.addContent(content) i iv iv iv iv iv iv iv iv iv</pre>	
	#21月4分中文 /** * MyToolMindow实用于哲理工用窗口指关的操作 * @param toolMindow 工目窗口时象	<pre>35 addActionListener { 66 label.text = MyBundle.message("randomLabel", service.getRandomNumber()) 61 } 63 } 64 }) 65 }</pre>	
	*/ class MyToolWindow(toolWindow: ToolWindow) { // 供用工具的口对培治Dproject居性共取项目服务 private val service = toolWindow.project.service <myprojectservice>() /** * 就配工目前口内容 * @return 该回一个10Pranel对操作为工具密口的内容</myprojectservice>		

15.8 解释代码

• 解释代码(选中代码段后右键,后续操作同理)

-	I Intellij-platform-plugin-template-main ~ 能非担例 ~		
		Gn MyToolWindowFactory.kt ×	
80 	Come and experience efficient code completion! Come and experience efficient code completion function. • Press tab to accept all completion suggestions. • Press Ctri+ to accept a single word completion suggestions. • Press Ctri+ to accept a single word completion suggestion. • Press Ctri+ to accept a single word completion suggestion. • Press Ctri+ to accept a single word completion suggestion. • Press Ctri+ to accept a single word completion suggestion. • Press Ctri+ to accept a single word completion suggestion. • Press Ctri+ to accept a single word completion suggestion.	<pre>package org.jetbrains.plugins.template.toolWindow } package org.jetbrains.plugins.template.toolWindow } inport i</pre>	
	If there are any Fitten Code related issues or suggestions, Welcome to click here for feedback	<pre>22 22 22 23 24 25 25 25 25 25 25 25 25 25 25 25 25 25</pre>	 Friten Code - Explain Code Filten Code - Start Chat Filten Code - Start Chat Filten Code - Generate Code Filten Code - Generate Unit Test Filten Code - Edit Code

15.8 解释代码

• 解释代码(选中代码段后右键,后续操作同理)

-	🗮 🛤 intellij-platfor				
			G MyToolWindowFactory	yat -	
	图 Explain Code	G X	package org.je	jetbrains.plugins.template.toolWindow	
80 	The provided code snipps override function to cre Let's break down the cod 1. override fun creat This line declares a fun ToolkindowFactory with the project and to	et is from a Kotlin file, likely within an Intellid IDEA plugin, and it contains an ate the content of a tool window within the IDE. e step by step: teToolMindowContent(project: Project, toolMindow: ToolWindow) nction createToolMindowContent which overrides a function defined in the interface. When a tool window is created within the IDE, this function is calle ool window as parameters.	14 15 14 15 15 17 17 17 17 10 11 17 10 11 15 10 10 10 10 10 10 10 10 10 10	WindowFectory : ToolWindowFectory { Logger().warn("Son't forget to remove all non-needed sample code files with their corresponding registration ent	
	2 val syToolxindow of Here, a new instance of parameters to its const 3 val content = ContextFactory.get false) This line creates the or instance using the cord toolxindow content Finally, the newly creates in summary, the main function and provide the adding its content on the badding its content on the specific content and behavior.	• MyToolkindow(toolkindow) of the MyToolkindow class is created, passing the toolkindow from the fun tructor. CInstance().createContent(myToolkindow.getContent(), null, antent for the tool window. It uses the ContentFactory to create a content ternt of the MyToolkindow instance created in the previous step. Otanager_addContent(content) ted content is added to the contentHanager of the toolkindow. ction of this code snippet is to override the createToolkindow. content for the tool window by creating an instance of MyToolkindow and tool window's content manager. The MyToolkindow class likely defines the wind of the tool window.	21 22 everride 1 23 val m 24 val m 25 toolm 27 28 override 1 27 29 override 1 27 29 override 1 27 29 override 1 27 29 override 1 20 overde 1 20 override 1 20 overide 1 20 override 1 20 override 1	<pre>fun createToolWindowContent(project: Project, toolWindow: ToolWindow) { eyToolWindow(toolWindow) content = ContentFactory.getInstance().createContent(myToolWindow.getContent(), null, false) #indow.contentManager.addContent(content) run shouldBeAvailable(project: Project) = true ToolWindow(toolWindow: ToolWindow) { ste wal service = toolWindow.project.service=RyProjectService>() </pre>	
	www.idef0作用品什么 overridef0作用品什么 override天健子用于在 在結正的分前体子服供一个 按口时被重写。这种转着 面下的作用体子服供一个 性、子关可以定制化后为。 在 Kotlin中,使用 overri 实现,并且可以提高代码 kepty.	可能設備同 「教生要提问 「教生要提问 「教生要集」 文字或是日中已定义的函数或属性。 de 美雄字表明 createToolWindowContent 函数指型加速程でのWindowFacto 百貨特定実中、createToolWindowContent 函数指置加速度日中的函数成 面積特定実中、createToolWindowContent 函数指置加速度日中的函数成 面積特定実中、createToolWindowContent 函数指置加速度目中的函数成 如具符合特定的上下文或需求。 de 美雄字来單写文类或能口中已定义的函数或属性,有助于理解代码的毛術 的可能性和可能的性。	11. 14. fun gr 15. vv 16. 17. ar 11. ar	<pre>getContent() = JBPanel<jbpanel<*>>().apply { val label = JBLabel(MyBundle.message("randomLabel", "?")) add(label) add(JButton(MyBundle.message("shuffle")).apply { addActionListener { label.text = MyBundle.message("randomLabel", service.getRandomNumber()) } })</jbpanel<*></pre>	

15.9 生成测试

• 生成单元测试

1000

-	19 🗮 🔰 Intellij-platform-plugin-template-main 🗸 版本控制 🗸					
		G MyToolWindowFactory.kt ×				
80	Come and experience efficient code completion! Open the code file you are writing and enter any code to use the auto completion function. • Press tab to accept all completion suggestions. • Press Ctrl+ II to accept a line of completion suggestions. • Press Ctrl+ II to accept a single word completion suggestion. You can also click on the "chat icon II" above to experience practical small features. For example click on the chat icon above to start chatting with the AI code assistant. If there are any Fitten Code related issues or suggestions. Welcome to click here for feedback	<pre>package org.jetbrains.plugins.template.toolWindow package org.jetbrains.plugins.template.toolWindow class MyToolWindowFactory : ToolWindowFactory { init { thisLogger().warn("Don't forget to remove all non-needed sample code files with their corresponding registration en } override fun createToolWindowContent(project: Project, toolWindow: ToolWindow) { val myToolWindow = MyToolWindow(toolWindow) val content = ContentFactory.getInstance().createContent(myToolWindow.getContent(), null, false) toolWindow.contentManager.addContent(content) } override fun shouldBeAvailable(project: Project) = true </pre>				
		<pre>class MyToolWindow(toolWindow: ToolWindow) { private val service = toolWindow.project.service(H) fun getContent() = JBPanet<jbpanet<.pre>/ URLF文操作 AltsEnter % %U(L) Ctrix(\$ %U(L) C</jbpanet<.pre></pre>				

15.9 生成测试



📃 📫 intellij-platform-plugin-template-main 🗸 版本控制 🗸				
	@ MyToolWindowFactory.kt imes			
	1 package org.jetbrains.plugins.template.toolWindow			
占 Generate Unit Test	2 3 ⇒ jmport			
<pre>import com.intellij.testFramework.UsefulTestCase import org.junit.Test import org.mockito.kotlin.mock import org.mockito.kotlin.whenever class MyToolWindowTest { @Test fun testGetContent_happyPath() { val toolWindow = mock<toolwindow>() val projectService = mock<myprojectservice>() val myToolWindow = MyToolWindow(toolWindow) whenever(toolWindow.project.service<myprojectservice>()).thenReturn(whenever(toolWindow.project.service<myprojectservice>()).thenReturn(whenever(toolWindow.project.service<myprojectservice>()).thenReturn(</myprojectservice></myprojectservice></myprojectservice></myprojectservice></toolwindow></pre>	<pre>init { class MyToolWindowFactory : ToolWindowFactory { init { thisLogger().warn(*Don't forget to remove all non-needed sample code files with their corresponding registration entries in `plugin.xml`.") } override fun createToolWindowContent(project: Project, toolWindow: ToolWindow) { val myToolWindow = MyToolWindow(toolWindow) val content = ContentFactory.getInstance().createContent(myToolWindow.getContent(), null, false) toolWindow.contentManager.addContent(content) } override fun shouldBeAvailable(project: Project) = true</pre>			
	29			
<pre>val content = myToolWindow.getContent() // Assert content contains a label and a button UsefulTestCase.assertNotEmpty(content.components) } WTest</pre>	30 class MyToolWindow(toolWindow: ToolWindow) { 31 32 32 private val service = toolWindow.project.service <myprojectservice>() 33 </myprojectservice>			
<pre>fun testGetContent_edgeCaseEmptyMessage() {</pre>	36			
<pre>val toolwindow = mock(solwindow>() val projectService = mock(MyProjectService>() </pre>	37 add(label) 38 add(JButton(NyBundle.message("shuffle")).apply { 39 addActionListemer {			
Val myrodiwindow = Myrodiwindow(todiwindow)	<pre>40 label.text = MyBundle.message("randomLabel", service.getRandomNumber()) </pre>			
<pre>whenever(toolWindow.project.service</pre> // thenReturn(whenever(projectService.getRandomNumber()).thenReturn()	$\begin{array}{c} a_1 \\ a_2 \\ a_3 \end{array}$			
<pre>val content = myToolWindow.getContent()</pre>				
<pre>// Assert label is still created but with an empty message vel label = content.getComponents(J8Label::class.jeva).firstOrNull() UsefulTestCase.assertNotNull(label) UsefulTestCase.assertEquals("", label.text) } } Ccopy </pre>				

15.10 检查 BUG

• 查找bug

	🕲 🗮 į inteliij-piatform-plugin-template-main 🗸 版本控制 🗸					
\Box		💮 МуТ	oolWindowFactory.kt ×			
80 	Come and experience efficient code completion! Open the code file you are writing and enter any code to use the auto completion function. Press tab to accept all completion suggestions. Press Ctrl+ To accept a line of completion suggestions. Press Ctrl+ To accept a single word completion suggestion. You can also click on the "chat icon		<pre>package org.jetbrains.plugins.template.toolWindow import class MyToolWindowFactory : ToolWindowFactory { init { thisLogger().warn("Don't forget to remove all non-needed sample code files with their }</pre>		stration entri	
	If there are any Fitten Code related issues or suggestions, Welcome to click here for feedback		<pre>override fun createToolWindowContent(project: Project, toolWindow: ToolWindow) { vel myToolWindow = MyToolWindow(toolWindow) val content = ContentFactory.getInstance().createContent(myToolWindow.getContent(), m toolWindow.contentHanager.addContent(content) } override fun shouldBeAvailable(project: Project) = true claws MyToolWindow(toolWindow: ToolWindow) {</pre>	Fitten Code ♀ 显示上下文操作 氷 剪切(T) 回 反相(C) ④ 約期(P) 反相(活動約% 別(254約=00)		Fitten Code - Explain Code Fitten Code - Start Chat Fitten Code - Find Bugs Fitten Code - Generate Code Fitten Code - Generate Unit Test Fitten Code - Document Code Fitten Code - Edit Code
			<pre>dlass myrootkindok(totkindok. rootkindok) { private val service = toolWindow.project.service<myprojectservice>() fun getContent() = JBPanel<jbpanel<*>>().apply { val label = JBLabel(MyBundle.message("randomLabel", "?")) add(label) add(JButton(MyBundle.message("shuffle")).apply { } }</jbpanel<*></myprojectservice></pre>			
			addActionListener {	旅時(R) 生成 打开于 Local <u>History</u> (家 与那帖校比较(<u>B</u>)		

15.10 检查 BUG

• 查找bug

三 🕅 Intellij-platform-plugin-template-main 🗸 R本控制 🗸				
	i MyToolWindowFactory.kt ×			
ය X ව වී Find Bugs	package arg.jetbrains.plugins.template.toolWindow > import			
The provided code appears to be a Kotlin function used to create the content for a tool window in an IntelliJ IDEA plugin. Potential issues in the code could include: 1. NullPointerException: The HyToolWindow or toolWindow object might be null, in which case calling methods on them could result in a NullPointerException. 2. Potential memory leaks: If the Content object created by ContentFactory.getInstance().createContent is not properly managed (i.e., removed when it's no longer needed), it could lead to memory leaks. 3. Error handling: There is no error handling in the code snippet provided. If any of the method colls or block methods for the inclusion of the method colls or block methods for the inclusion.	<pre>14 15 16 17 17 18 19 19 19 10 10 10 11 17 10 10 11 1 10 10 10 10 10 10 10 10 10 1</pre>			
cails or object creations tails, there is no error nanoting or logging to indicate the problem. To address potential assies, you can perform the following:	22 override fun createToolWindowContent(project: Project, toolWindow: ToolWindow) { 21 val myToolWindow = MyToolWindow(toolWindow) 22 val myToolWindow = MyToolWindow(toolWindow)			
<pre>override fun createToolWindowContent(project: Project, toolWindow: ToolWindow) if (project != null && toolWindow != null) { val myToolWindow = MyToolWindow(toolWindow) val content + ContentFactory.getInstance().createContent(myToolWindow. if (content != null) { toolWindow.contentManager.addContent(content) } else { // Handle the case when content creation fails } }</pre>	<pre>table content = contentPatter getIntable() created inter(syndocandousgetEnter(), net(, net(), net(, net(), ne</pre>			
<pre>} } else { // Handle the case when project or toolkindow is null } } Compy</pre>	<pre>sz private val service = toolWindow.project.service<myprojectservice>() s fun getContent() = JBPanel<jbpenel<*>().apply { val label = JBLabel(MyBundle.message(*randomLabel*, *?*)) }</jbpenel<*></myprojectservice></pre>			
In this modified code snippet, the nullability of project and toolaindow is explicitly checked, and proper error handling is added for the content creation. This helps to mitigate potential issues related to null pointers and content creation failures.	add(Label) add(JButton(MyBundle.message("shuffle")).apply { addActionListener { label.text = MyBundle.message("randomlabel", service.getRandomNumber())			

15.11 编辑代码

• 编辑代码

📃 📫 Inteltij-platform-plugin-template-main 🐃 1k # 12 M 😁					6₹ Run Tests ∽	D 0 1
	(in My	TeolWindowFactory.kt ×				
 G X D Come and experience efficient code completion! Open the code file you are writing and enter any code to use the auto completion function. • Press tab to accept all completion suggestions.		package org.jetbrains.plugins.template.toolWindow > import class MyTeelWindowFactory : TeelWindowFactory (
 Press Ctri+ et to accept a sine of completion suggestions. Press Ctri+ et to accept a single word completion suggestion. You can also click on the "chat icon et al. above to experience practical small features. For example, click on the chat icon above to tati chatting with the Al code assistant. 		<pre>init { thisLogger().warn('Don't forget to remove all non-needed sample code files with their corresponding rec }</pre>				
If there are any Fitten Code related issues or suggestions, Welcome to click here for feedback		<pre>override tur prestelogizindoxContent(project: Project; tool#indox: Tool#indox) { vel myTool#indox = myTool#indox(tool#indox) vel content = ContentFactory.getInstance().createContent(myTool#indox.getContent(), null, false) tool#indox.contentHanager.addCentent(content) } override fun shouldBeAvailable(project: Project) = true class MyTool#indox(tool#indox: Tool#indox) { private val service = tool#indox.project.service<myprojectservice>() fun getContent() = JBPanel<jbpanel<ad>().spply { val label = JBLabel(MyBundle.nessage("rendomLabel", ")")) sdd(label) sdd(label) sdd(label) sdd(stort = f label.text = MyBundle.message("rendomLabel", service.getRendomWumber()) } } } </jbpanel<ad></myprojectservice></pre>	Fitten Code 早 日本上下之操作 学 打切(工) 御 友知(四) 御 友知(四) 御 友知(四) 御 友知(四) 章 友知(四) 常政 市政 秋雨 北雨 日本小雨 上成… 打声子 上ocal History 学 均衡雨疾北秋雨)	X VIT-Enter Curr+X Curr+G Curr+G Curr+G Curr+G Att+Shift+Insert Att+Shift+Insert Att+Shift+Insert Att+Shift+Insert XIT+Shift+In	Ritten Code - Explain Code Ritten Code - Start Chat Ritten Code - Start Chat Ritten Code - Generate Code Ritten Code - Generate Unit Test Ritten Code - Document Code Ritten Code - Edit Code	

15.11 编辑代码

• 编辑代码

🜒 🗮 📫 intellij-platform-plugin-template-main 🐃 🕸 A (2) M 🛩	
- i -	t MyToolWindowFactory.kt ×
G ≯ ₽ ℓ Edit Code	package org.jetbrains.plugins.template.toolWindow
<pre>00 \$\$7\$paveftB, MEDGUD+文計算 @Override public void createToolWindowContent(Project project, ToolWindow toolWindow) { // Optice文工具由工具的 MyToolWindow eyToolWindow = new MyToolWindow(toolWindow); // Optice文工具由工具的 MyToolWindow = new MyToolWindow(toolWindow); // Optice文工具由工具的 MyToolWindow = new MyToolWindow(toolWindow); // Optice文工具由工具的 Content content = ContentFactory.getInstance().createContent(myToolWindow.</pre>	<pre>14 15 16 17 18 19 19 19 19 19 19 19 10 10 10 10 10 10 10 10 10 10 10 10 10</pre>
toolWindow.getContentManager().addContent(content); } Copy	22 averride fun createToolWindowContent(project: Project, toolWindow: ToolWindow) { 23 val myToolWindow = MyToolWindow(toolWindow) 24 val.content = ContentFactory.getInstance().createContent(myToolWindow.getContent(), null, false) 25 toolWindow.sontentHanager.addContent(content)
Redy	<pre>averride fun shouldBeAvailable(project: Project) = true class MyToolWindow(toolWindow: ToolWindow) { privete val service = toolWindow.project.service<myprojectservice>() fun getCentent() = JBPanel(JBPanel(JBPanel(JBPanel(JBPanel(JBPanel(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--/-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->(JBPanel<!--//-->/</myprojectservice></pre>

15.12 常见问题

- 点击右下角Fitten图标,然后选择"打开插件设置",即可打开Fitten Code的设置页面。以下 为可设置选项
- 1. 延迟自动补全的时间(单位:毫秒)
- 2. Fitten Code的显示语言
- 3. 使用"Fitten Code 生成注释"功能时的语言首选项

15.12 常见问题

- 12.1 IDEA低版本搜不到Fitten Code
- 答: 2024/3/21号发布了0.10.3版本,最低支持2021版本JetBrain。
- 12.2 Android Studio侧边栏没有反应
- 答: 安装JCEF, 并且关闭sandbox即可。
- 12.3 idea英文但是想让fitten是中文的方便看代码解释
- 答:点击右下角Fitten图标,然后选择"打开插件设置",即可设置Fitten Code的语言为中文。
- 12.4 如何选择各功能回答时的语言?
- 答:点击右下角Fitten图标,然后选择"打开插件设置",即可设置对应功能回答的语言。



•掌握以下内容: Fitten Code安装 使用Fitten Code对话的方式生成代码 <u>میازلی</u>م ×⊢ 使用Fitten Code查找BUG Ο × ٥ 鼏 0

本章作业

- 完成以下内容:
- 1 IntelliJ IDEA 中安装Fitten Code组件
- 2 使用Fitten Code对话的方式生成代码
- 3 使用Fitten Code查找BUG





