

Child Bank Program

Idea 项目配置 (基于Maven的项目搭建)

- pom.xml的配置 (渲染 + JUnit配置)

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.example</groupId>
  <artifactId>childBank</artifactId>
  <version>1.0-SNAPSHOT</version>
  <name>Archetype - childBank</name>
  <url>http://maven.apache.org</url>

  <properties>
    <!-- 定义Junit Jupiter的版本号 -->
    <junit-jupiter.version>5.9.2</junit-jupiter.version>
    <!-- 其他可能的属性定义 -->
  </properties>

  <dependencies>
    <dependency>
      <groupId>com.formdev</groupId>
      <artifactId>flatlaf</artifactId>
      <version>3.4</version>
    </dependency>

    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter</artifactId>
      <version>${junit-jupiter.version}</version>
      <scope>test</scope>
    </dependency>

  </dependencies>

</project>
```

- 文件结构搭建 (外加JUnit部分搭建)

```
D:\.
├─main
│   └─java
│       │   mainwindow.form
│       │   mainwindow.java
│       │
│       └─child
│           ChildAcc.java
```

```

| | |
| | └─UI_page
| |     mainPage.java
| |
| └─resources
|     └─archetype-resources
|         | | pom.xml
|         | |
|         └─src
|             | └─main
|             |     └─java
|             |         App.java
|             |
|             └─test
|                 └─java
|                     AppTest.java
|
| └─META-INF
|     └─maven
|         archetype.xml
|
└─test
    └─java
        TestMainUI.java

```

- 测试部分代码示例

```

import child.ChildAcc;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.assertEquals;

public class TestMainUI {

    @Test
    public void test01(){
        ChildAcc child = new ChildAcc("Name");
        String stuName = child.getName();
        assertEquals(stuName, "Name");
    }
}

```

- GUI渲染部分代码示例

```

public class mainPage {
    public static void main(String[] args) {

        // 渲染界面风格：暗色调 / 亮色调
        // FlatDarculaLaf.setup();
        FlatLightLaf.setup();

        // 创建中间容器（面板容器）
        JPanel panel = new JPanel();

        // 创建一个基本组件（按钮），并添加到 面板容器 中
        JButton btn = new JButton("测试按钮");
    }
}

```

```

panel.add(btn);

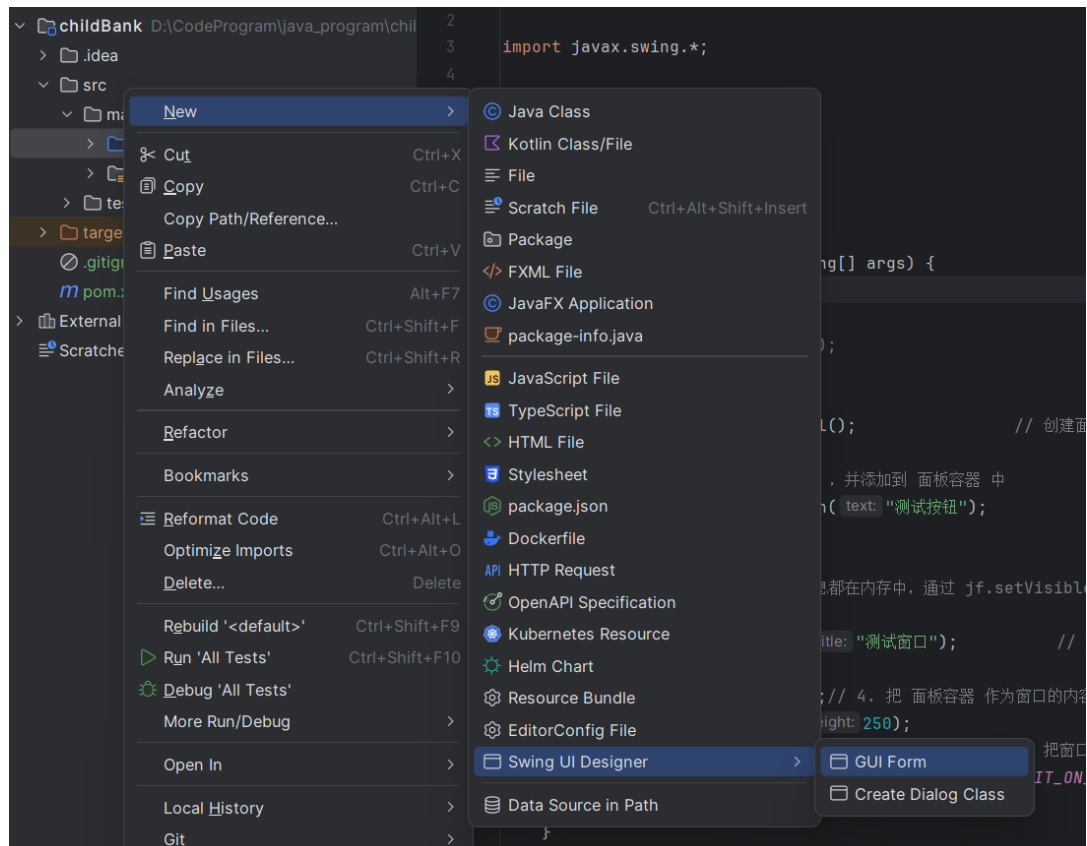
// 显示窗口，前面创建的信息都在内存中，通过 jf.setVisible(true) 把内存中的窗口
// 显示在屏幕上。
// 创建一个顶层容器（窗口）
JFrame jf = new JFrame("测试窗口");

// 把 面板容器 作为窗口的内容面板 设置到 窗口
jf.setContentPane(panel);
jf.setSize(250, 250); // 设置窗口大小
jf.setLocationRelativeTo(null); // 把窗口位置设置到屏幕中心
// 当点击窗口的关闭按钮时退出程序（没有这一句，程序不会退出）
jf.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
jf.setVisible(true);
}
}

```

- IDEA默认Swing界面可视化搭建

1. 在main的java文件夹下右键，如下图点击创建



2. 直接输入名称，进行创建



New GUI Form



Form name:



Base layout manager:

GridLayoutManager (IntelliJ)



Create bound class

Class name:

OK

Cancel