**Руководство для программиста по приложению "ConvertPlus" (Конвертер валют)**

Это руководство предназначено для разработчиков, которые хотят понять, как работает приложение "ConvertPlus", а также для тех, кто планирует вносить изменения или улучшения в код. Приложение написано на Python с использованием библиотеки PyQt5 и SQLite для хранения данных о валютных курсах.

**Основные компоненты приложения**

**1. Импорт необходимых библиотек**

* Приложение использует следующие библиотеки:
* PyQt5: для создания графического интерфейса пользователя (GUI).
* decimal: для точных вычислений с плавающей запятой, что особенно важно при работе с денежными суммами.
* sqlite3: для взаимодействия с базой данных SQLite, где хранятся курсы валют.

**2. Класс MainWindow**

Класс MainWindow наследует от QMainWindow и реализует основной интерфейс приложения.

Конструктор “\_\_init\_\_”

В конструкторе выполняются следующие действия:

* Установка заголовка окна.
* Загрузка доступных валют из базы данных.
* Установка валидаторов для текстовых полей ввода, чтобы разрешить ввод только чисел с плавающей точкой и двумя знаками после запятой.
* Инициализация полей ввода значениями по умолчанию (0).
* Подключение сигналов к методам конвертации и обмена значениями.

**3. Метод load\_currencies**

Этот метод загружает доступные валюты и их курсы из базы данных SQLite:

def load\_currencies(self):  
 try:  
 conn = sqlite3.connect('currency\_rates.db')  
 cursor = conn.cursor()  
 cursor.execute("SELECT currency\_code, rate FROM currency\_rates")  
 rows = cursor.fetchall()  
 self.currencies = {currency: rate for currency, rate in rows}  
 currencies = list(self.currencies.keys())  
 self.Sum1BOX.addItems(currencies)  
 self.Sum2BOX.addItems(currencies)  
 conn.close()  
 except Exception as e:  
 print(f"Ошибка при загрузке валют: {str(e)}")

**4. Методы конвертации**

“converter\_from\_sum1”

Этот метод выполняет конвертацию суммы из первого поля во второе:

def converter\_from\_sum1(self):  
 try:  
 amount = decimal.Decimal(self.Sum1.text()) if self.Sum1.text() else decimal.Decimal(0)  
 from\_currency = self.Sum1BOX.currentText() # Выбранная валюта  
 to\_currency = self.Sum2BOX.currentText() # Выбранная валюта  
  
 if from\_currency in self.currencies and to\_currency in self.currencies:  
 from\_rate = decimal.Decimal(self.currencies[from\_currency])  
 to\_rate = decimal.Decimal(self.currencies[to\_currency])  
 converted\_amount = (amount / from\_rate) \* to\_rate  
  
 # Округляем до двух знаков после десятичной точки  
 converted\_amount = converted\_amount.quantize(decimal.Decimal('1.00'))  
  
 # Удаляем ненужные нули после десятичной точки  
 converted\_amount = converted\_amount.normalize()  
  
 self.Sum2.setText(f"{converted\_amount}")  
 else:  
 self.Sum2.setText("Ошибка: валюта не найдена")  
 except ValueError:  
 self.Sum2.setText("Ошибка: неверный ввод")  
 except Exception as e:  
 self.Sum2.setText(f"Ошибка: {str(e)}")

“converter\_from\_sum2”

Аналогично, этот метод выполняет конвертацию суммы из второго поля в первое:

def converter\_from\_sum2(self):  
 try:  
 amount = decimal.Decimal(self.Sum2.text()) if self.Sum2.text() else decimal.Decimal(0)  
 from\_currency = self.Sum2BOX.currentText() # Выбранная валюта  
 to\_currency = self.Sum1BOX.currentText() # Выбранная валюта  
  
 if from\_currency in self.currencies and to\_currency in self.currencies:  
 from\_rate = decimal.Decimal(self.currencies[from\_currency])  
 to\_rate = decimal.Decimal(self.currencies[to\_currency])  
 converted\_amount = (amount / from\_rate) \* to\_rate  
  
 # Округляем до двух знаков после десятичной точки  
 converted\_amount = converted\_amount.quantize(decimal.Decimal('1.00'))  
  
 # Удаляем ненужные нули после десятичной точки  
 converted\_amount = converted\_amount.normalize()  
  
 self.Sum1.setText(f"{converted\_amount}")  
 else:  
 self.Sum1.setText("Ошибка: валюта не найдена")  
 except ValueError:  
 self.Sum1.setText("Ошибка: неверный ввод")  
 except Exception as e:  
 self.Sum1.setText(f"Ошибка: {str(e)}")

**5. Метод “swap\_values”**

Этот метод меняет местами выбранные валюты и очищает поля ввода:

def swap\_values(self):  
 sum1box\_value = self.Sum1BOX.currentText()  
 sum2box\_value = self.Sum2BOX.currentText()  
  
 self.Sum1BOX.setCurrentText(sum2box\_value)  
 self.Sum2BOX.setCurrentText(sum1box\_value)  
  
 self.Sum1.setText("0")  
 self.Sum2.setText("0")

**Запуск приложения**

Для запуска приложения необходимо выполнить следующий код:

if \_\_name\_\_ == '\_\_main\_\_':  
 app = QApplication(sys.argv)  
 window = MainWindow()  
 window.show()  
 sys.exit(app.exec\_())