**Объектно-ориентированные возможности языка Python.**

Выполнил: Шебеко Николай  
Группа: ИУ5-33Б  
Дата: 10.12.24г.

Описание задания:

Сделайть тг бота, который будет переводить текст с помощью яндекс переводчика

Программный код:

from telegram import Update, ReplyKeyboardMarkup  
from telegram.ext import Updater, CommandHandler, MessageHandler, Filters, CallbackContext  
import requests  
  
YANDEX\_API\_KEY = 'yandex\_translate\_api\_key'  
YANDEX\_API\_URL = 'https://translate.yandex.net/api/v1.5/tr.json/translate'  
  
target\_languages = {}  
  
def start(update: Update, context: CallbackContext) -> None:  
 user = update.effective\_user  
 target\_languages[user.id] = 'en'  
 update.message.reply\_text(  
 f"Привет, {user.first\_name}! Я бот для перевода текста.\n"  
 "Отправь мне текст для перевода, и я переведу его на английский.\n"  
 "Используй /setlang, чтобы выбрать целевой язык."  
 )  
  
def help\_command(update: Update, context: CallbackContext) -> None:  
 update.message.reply\_text(  
 "Я бот для перевода текста. Вот что я могу:\n"  
 "- Отправь мне текст, и я переведу его на выбранный язык.\n"  
 "- Используй /setlang, чтобы изменить целевой язык.\n"  
 "Поддерживаемые языки: en (английский), ru (русский), es (испанский), и другие."  
 )  
  
def set\_language(update: Update, context: CallbackContext) -> None:  
 user = update.effective\_user  
 if context.args:  
 lang = context.args[0]  
 target\_languages[user.id] = lang  
 update.message.reply\_text(f"Целевой язык установлен: {lang}")  
 else:  
 update.message.reply\_text("Пожалуйста, укажите язык. Например: /setlang en")  
  
def translate\_text(update: Update, context: CallbackContext) -> None:  
 user = update.effective\_user  
 text\_to\_translate = update.message.text  
 target\_language = target\_languages.get(user.id, 'en')  
 params = {'key': YANDEX\_API\_KEY, 'text': text\_to\_translate, 'lang': target\_language}  
 response = requests.get(YANDEX\_API\_URL, params=params)  
 data = response.json()  
  
 if response.status\_code == 200 and 'text' in data:  
 translated\_text = data['text'][0]  
 update.message.reply\_text(f"Перевод ({target\_language}): {translated\_text}")  
 else:  
 update.message.reply\_text("Произошла ошибка при переводе текста. Попробуйте позже.")  
  
def main() -> None:  
 TELEGRAM\_BOT\_TOKEN = 'telegram\_bot\_token'  
 updater = Updater(TELEGRAM\_BOT\_TOKEN)  
 dispatcher = updater.dispatcher  
 dispatcher.add\_handler(CommandHandler("start", start))  
 dispatcher.add\_handler(CommandHandler("help", help\_command))  
 dispatcher.add\_handler(CommandHandler("setlang", set\_language))  
 dispatcher.add\_handler(MessageHandler(Filters.text & ~Filters.command, translate\_text))  
 updater.start\_polling()  
 updater.idle()  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()