9B

Popište základní typy uživatelských rozhraní (CLI, GUI, webové). Popište nástroje používané v jazyku Python pro tvorbu rozhraní (click, Tkinter, Qt, Flask, Django).

CLI (Command Line Interface)

Uživatelské rozhraní ve formě příkazového řádku, ikonické pro Linux. Dostupné na všech systémech ve formě terminalu, nebo cmd. Pohyb v systému přes příkazy (cd, ls, nano, ...).

Více náročné na naučení potřebných příkazů, ale mnohem rychlejší než GUI

GUI (Graphical User Interface)

Nejpoužívanější rozhraní v systémech. Vše je graficky zobrazeno, ale je k dispozici i možnost CLI. Je snadnější se v něm pohybovat, obzvlášť pro běžné uživatele ale kvůli potřebě proklikávat se přes různá okna je pomalejší než CLI.

Webové rozhraní

Rozhraní nějaké aplikace, stránky, či celého systému přes webovou aplikaci. Dá se využít rozhraní CLI, ale opět je nejpoužívanější GUI. Neni nutnost žádné instalace, slouží jako alternativní přístup. Příklady můžou být různé online nástroje, připojení přes web v Haxagonu, nebo správa webové stránky přes CMS.

Click (Command Line Interface Creation Kit)

Package jazyku Python pro tvorbu CLI rozhraní. Snaha o co největší zjednodušení tvorby příkazového rozhraní pro vývojáře.

**import** click  
  
**@click.**command**()**  
**@click.**option**(**'--count'**,** default**=1,** help**=**'Number of greetings.'**)**  
**@click.**option**(**'--name'**,** prompt**=**'Your name'**,**  
 help**=**'The person to greet.'**)**  
**def** hello**(**count**,** name**):**  
 *"""Simple program that greets NAME for a total of COUNT times."""*  
 **for** x **in** range**(**count**):**  
 click**.**echo**(**f"Hello {name}!"**)**  
  
**if** \_\_name\_\_ **==** '\_\_main\_\_'**:**  
 hello**()**

Tkinter

Package pro Python, umožňující vytváření grafického rozhraní. Zaručená funkčnost na distribucích Linuxu, ve Windows a macOS. Jednoduché vytváření oken aplikací a následného rozhraní.

Qt

Stejně jako Tkinter slouží pro vytváření grafického rozhraní. Má k dispozici dvě možnosti tvorby: Qt Widgets a Qt Quick.

Widgets:

**class** **MyWidget**(QtWidgets.QWidget):  
 **def** \_\_init\_\_(self):  
 super().\_\_init\_\_()  
  
 self.hello = ["Hallo Welt", "Hei maailma", "Hola Mundo", "Привет мир"]  
  
 self.button = QtWidgets.QPushButton("Click me!")  
 self.text = QtWidgets.QLabel("Hello World",  
 alignment=QtCore.Qt.AlignCenter)  
  
 self.layout = QtWidgets.QVBoxLayout(self)  
 self.layout.addWidget(self.text)  
 self.layout.addWidget(self.button)  
  
 self.button.clicked.connect(self.magic)  
  
 **@QtCore**.Slot()  
 **def** magic(self):  
 self.text.setText(random.choice(self.hello))

Quick:

**import** QtQuick  
**import** QtQuick.Controls  
**import** QtQuick.Layouts  
  
Window {  
 width: 300  
 height: 200  
 visible: **true**  
 title: "Hello World"  
  
 readonly property list<string> texts: ["Hallo Welt", "Hei maailma",  
 "Hola Mundo", "Привет мир"]  
  
 **function** setText() {  
 **var** i = Math.round(Math.random() \* 3)  
 text.text = texts[i]  
 }  
  
 ColumnLayout {  
 anchors.fill: parent  
  
 Text {  
 id: text  
 text: "Hello World"  
 Layout.alignment: Qt.AlignHCenter  
 }  
 Button {  
 text: "Click me"  
 Layout.alignment: Qt.AlignHCenter  
 onClicked: setText()  
 }  
 }  
}

Flask

Package pro vytváření webového rozhraní. Zaměření najednoduché vytvoření webové aplikace v jediném python souboru. Používá jej např. Netflix. Podpora rozšíření od ostatních vývojářů.

Django

Stejně jako Flask, slouží k vytváření webového rozhraní. Semi-rival Flasku. Dá se použít bez databáze, ale má vestavěnou funkci implementace databáze.