**CHECKPOINT 2**

|  |  |
| --- | --- |
| **Bộ môn:** Coding | **Năm môn:** App Producer - Intensive |

# 

# **TRẮC NGHIỆM (5đ)**

|  |  |
| --- | --- |
| **CÂU HỎI 1:** | Tại sao cần sử dụng định dạng JSON? |
| A. | Vì JSON là một ngôn ngữ lập trình chuyên dụng, giúp tối ưu hóa hiệu suất hệ thống. |
| B. | Do JSON có cấu trúc dữ liệu phức tạp, giúp bảo vệ thông tin quan trọng. |
| C. | JSON là một định dạng dữ liệu nhẹ và dễ đọc, thích hợp cho việc truyền tải thông tin giữa các ứng dụng. |
| D. | Vì JSON là ngôn ngữ dựa trên JavaScript, giúp tương thích tốt với nhiều ứng dụng web hiện đại. |
| Đáp án đúng là: C | |
| **CÂU HỎI 2:** | Lệnh nào sau đây có tác dụng đọc file JSON trong Python? |
| A. | readFile("data.json") |
| B. | loadJSON("data.json") |
| C. | parseJSON("data.json") |
| D. | json.load(open("data.json") |
| Đáp án đúng là: D | |
| **CÂU HỎI 3:** | Khi nào cần thực hiện hàm ghi file JSON?  Chọn đáp án đúng nhất. |
| A. | Khi cần đọc dữ liệu từ chỉ một file JSON. |
| B. | Mỗi khi danh sách đối tượng có thay đổi. |
| C. | Khi muốn thay đổi nội dung của danh sách đối tượng. |
| D. | Khi chỉ cần đọc dữ liệu mà không cần ghi lại. |
| Đáp án đúng là: B | |
| **CÂU HỎI 4:** | Trong PyQt6, để thực hiện các thao tác thêm, xoá và sửa đổi dữ liệu trong một QListWidget, bạn sẽ sử dụng những phương thức nào? |
| A. | addItem, removeItem, editItem |
| B. | append, delete, modify |
| C. | insertItem, takeItem, setItem |
| D. | addEntry, removeEntry, updateEntry |
| Đáp án đúng là: C | |
| **CÂU HỎI 5:** | Để hiển thị danh sách các phần tử từ file dữ liệu lên giao diện ứng dụng. Bạn sẽ thực hiện các bước nào sau đây để đạt được mục tiêu này?" |
| A. | Kết nối với file dữ liệu, truy vấn dữ liệu và thêm từng mục vào QTableWidget bằng phương thức addItem. |
| B. | Kết nối với file dữ liệu, truy vấn dữ liệu và thêm từng mục vào QListWidget bằng phương thức addItem. |
| C. | Chỉ cần tạo một QTableWidget và sau đó thêm các mục một cách bình thường với addEntry để hiển thị dữ liệu từ cơ sở dữ liệu. |
| D. | Chỉ cần tạo một QListWidget và sau đó thêm các mục một cách bình thường với addEntry để hiển thị dữ liệu từ cơ sở dữ liệu. |
| Đáp án đúng là: B | |

|  |  |
| --- | --- |
| **CÂU HỎI 6:** | Đoạn code sau có vấn đề gì?  class MyWindow(QWidget):  def \_\_init\_\_(self):  super().\_\_init\_\_()  self.initUI()  def initUI(self):  # Tạo QListWidget  self.list\_widget = QListWidget(self)  # Tạo danh sách đối tượng  person\_list = [  Person("John Doe", 25),  Person("Jane Smith", 30),  Person("Bob Johnson", 22)  ]  # Thêm đối tượng vào QListWidget  for person in person\_list:  item = QListWidgetItem(f"Name: {person.name}, Age: {person.age}")  self.list\_widget.addItem(item)  # Xóa phần tử John Doe  del person\_list[0] |
| A. | Phải sử dụng addItems thay vì addItem. |
| B. | Phải sử dụng lệnh pop thay vì lệnh del để xoá đối tượng. |
| C. | Xóa đối tượng trong danh sách đối tượng nhưng chưa xoá trong QListWidget. |
| D. | Không có vấn đề gì. |
| Đáp án đúng là: C | |

|  |  |
| --- | --- |
| **CÂU HỎI 7:** | Đoạn code sau có vấn đề gì?  def search(self):  search\_anime\_field = widgets.inputAnime.text()  if search\_anime\_field:  matched\_items = widgets.animeList.findItems(search\_anime\_field, Qt.MatchFlag.MatchContains)  for i in range(widgets.animeList.count()):  it = widgets.animeList.item(i)  it.setHidden(it not in matched\_items)  else:  for i in range(widgets.animeList.count()):  it = widgets.animeList.item(i)  it.setHidden(False) |
| A. | Chưa xử lý string search\_anime\_field. |
| B. | Phải sử dụng findItem thay vì findItems. |
| C. | Sai điều kiện tìm trong phương thức findItems. |
| D. | Không có vấn đề gì. |
| Đáp án đúng là: A | |

|  |  |
| --- | --- |
| **CÂU HỎI 8:** | Trong ứng dụng PyQt, khi nào sự kiện xảy ra làm thay đổi dữ liệu đối tượng trên ổ cứng? |
| A. | Sau khi nhấn nút xoá phần tử trong ứng dụng. |
| B. | Sau khi nhấn nút tắt ứng dụng. |
| C. | Sau khi nhấn nút tìm kiếm. |
| D. | Sau khi chuyển sang trang khác trong Stacked Widget. |
| Đáp án đúng là: A | |

|  |  |
| --- | --- |
| **CÂU HỎI 9:** | Lệnh nào sau đây SAI cú pháp để kết nối tín hiệu và chỗ giữ? |
| A. | button.clicked.connect(lambda: self.show\_message("Button clicked")) |
| B. | line\_edit.textChanged.connect(lambda: self.on\_text\_changed(line\_edit.text())) |
| C. | button.clicked.connect(self.show\_message()) |
| D. | button.clicked.connect(self.show\_message) |
| Đáp án đúng là: C | |

|  |  |
| --- | --- |
| **CÂU HỎI 10:** | Phát biểu nào sau đây đúng về định dạng JSON? |
| A. | JSON chỉ có thể biểu diễn dữ liệu đơn giản. |
| B. | Chỉ có con người mới đọc được JSON. |
| C. | JSON thay thế được hoàn toàn các định dạng dữ liệu khác. |
| D. | JSON không chỉ được sử dụng để phát triển ứng dụng Python. |
| Đáp án đúng là: D | |

# **THỰC HÀNH (5đ)**

|  |  |
| --- | --- |
| **ĐỀ BÀI:** | Với phần mềm quản lý bài tập về nhà ở buổi trước  Lớp **Homework** gồm các thuộc tính như sau:   * Thuộc tính **name** chỉ tên bài tập. Ví dụ: “lập trình”, “cắt dán”, “làm văn”,... * Thuộc tính **priority** chỉ mức độ quan trọng của bài tập, gồm các mức: **1** - thấp, **2** - trung bình, **3** - cao. * Thuộc tính **completed** chỉ trạng thái của bài tập: hoàn thành hoặc không hoàn thành.   Lớp quản lý **HomeworkList** gồm các phương thức như sau:   * Phương thức khởi tạo danh sách đối tượng. * Phương thức thêm phần tử vào danh sách.   **Code mẫu từ bài trước**  class Homework:  def \_\_init\_\_(self, name, priority, completed=False):  self.name = name  self.priority = priority  self.completed = completed  class HomeworkList:  def \_\_init\_\_(self):  self.items = []  def add\_item(self, item):  self.items.append(item)  def all\_completed(self):  completed = True  for item in self.items:  if item.completed == False:  completed = False  print(item.name)  if completed:  print("All finished!")  hw\_list = HomeworkList()  hw\_list.add\_item(Homework("Lap trình App Producer", 3))  hw\_list.add\_item(Homework("Lam văn", 2, True))  hw\_list.add\_item(Homework("Lap trình GameMaker", 3))  *Hãy tạo giao diện Dialog:*   * *Có List Widget hiển thị tên của bài tập.* * *Có nút Export Data, khi nhấn nút sẽ xuất danh sách đối tượng* ***hw\_list*** *ra file dữ liệu* ***new\_data.json****.*   *Kết quả mong đợi:* |
| **HƯỚNG DẪN:** | class Dialog(QDialog):  def \_\_init\_\_(self, hw\_list:HomeworkList):  super().\_\_init\_\_()  self.ui = uic.loadUi("homework\_ui.ui", self)  self.hw\_list = hw\_list  for item in self.hw\_list.items:  self.ui.listWidget.addItem(item.name)    self.ui.export\_Button.clicked.connect(self.export\_data)    def export\_data(self):  file = "new\_data.json"  data = []  for item in self.hw\_list.items:  data.append(item.\_\_dict\_\_)    with open(file, "w") as f:  json.dump(data, f, indent=4)  if \_\_name\_\_ == "\_\_main\_\_":  hw\_list = HomeworkList()  hw\_list.add\_item(Homework("Lap trinh App Producer", 3))  hw\_list.add\_item(Homework("Lam van", 2, True))  hw\_list.add\_item(Homework("Lap trinh GameMaker", 3))  app = QApplication(sys.argv)  window = Dialog(hw\_list)  window.show()  sys.exit(app.exec()) |