**Recapitulare**

1. Selecteaza afirmatiile adevarate referitoare la Design Patterns in Python:
2. Design pattern-urile reprezinta cod definit care ne ajuta sa rezolvam o problema comuna, putem sa il copiem si sa rezolvam problemele similare. F
3. Design Patterns reprezinta un set de “good practices”, dupa care se ghideaza scrierea de cod in OOP. A
4. Folosin un design pattern, putem sa scriem cod mai curat, mai flexibil, extensibil si decuplat. A
5. Exista 3 categorii de design patterns: creational, behavioural, structural. A
6. Decorator este un design pattern structural. A

**Raspunsuri corecte: b, c, e**

1. Selecteaza afirmatiile adevarate:
2. Un obiect este iterator daca are implementata metoda \_\_iter\_\_(). F
3. Un obiect este iterabil daca are implementate metodele \_\_iter\_\_() si \_\_next\_\_(). F
4. Un obiect este iterator daca are implementate metodele \_\_enter\_\_() si \_\_exit\_\_(). F
5. Un obiect este iterabil daca are implementata metoda \_\_iter\_\_(). A
6. Un obiect este iterator daca are implementata metoda \_\_iter\_\_() si \_\_next\_\_(). A

**Raspunsuri corecte: d, e**

1. Selecteaza afirmatiile adevarate:
2. Un generator este un nou tip de date din Python. F
3. Un generator este o functie care are cel putin o instructiune yield. A
4. Spre deosebire de instructiunea return, instructiunea yield incheie o functie in intregime. F
5. Spre deosebire de instructiunea return, instructiunea yield opreste functia salvand toate starile sale si mai tarziu continua de acolo la apeluri succesive. A
6. Generatorii sunt tot iteratori, si metodele \_\_iter\_\_() si \_\_next\_\_() sunt create in mod automat. A

**Raspunsuri corecte: b, d, e**

1. Selecteaza afirmatiile adevarate:
2. Un context manager este o functie noua in Python. F
3. Un context manager este un obiect care implementeaza metodele \_\_iter\_\_() si \_\_next\_\_(). F
4. Un context manager este un obiect care implementeaza metodele \_\_enter\_\_() si \_\_next\_\_(). F
5. Un context manager este un obiect care implementeaza metodele \_\_enter\_\_() si \_\_exit\_\_(). A
6. Metodele \_\_enter\_\_() si \_\_exit\_\_() sunt metode apelate automat cand folosim blocul with. A

**Raspunsuri corecte:**

1. Selecteaza afirmatiile adevarate:
2. Decoratorii sunt functii cu utilizare rara in Python F
3. Decoratorii sunt functii foarte importante in Python si reprezinta un tool puternic pentru a extinde comportamentul altor functii. A
4. Cand folosim decoratori, implementarea functiilor care sunt decorate trebuie schimbata. F
5. Avantajul decoratoriilor este ca extind functionalitatea functiilor fara sa le modifice implementarea. A
6. Pentru a invoca un decorator, folosim @. A

**Raspunsuri corecte: b, d, e**

1. Care este diferenta principala intre un iterator si un generator in Python?
2. Un iterator este o functie, in timp ce un generator este un obiect. F
3. Un iterator trebuie sa implementeze metoda \_\_iter\_\_(), in timp ce un generator trebuie sa implementeze metoda \_\_next\_\_(). F
4. Un iterator este un obiect, in timp ce un generator este o functie. A
5. Un iterator trebuie sa implementeze metodele \_\_iter\_\_() si \_\_next\_\_(), in timp ce in cazul generatorilor, aceste metode nu sunt folosite/implementate. F
6. Nu exista diferente semnificative intre un iterator si un generator. F

**Raspunsuri corecte: c**