Lab 6 (GUI Quiz)

أكمل الكود

السؤال 1

أكمل الجزء الناقص لعرض النافذة:

```
public class QuizGUI {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Quiz");
        frame.setSize(400,300);

        // أكمل هنا
        _____________;

        frame.setVisible(true);
    }
}
```

الإجابة:

```
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

السؤال 2

```
submitButton.addActionListener(e -> {
    String answer = _____________;
    selectionLabel.setText("Your Selection: " + answer);
});
```

الإجابة:

```
(String) comboBox.getSelectedItem()
```

أوجد الخطأ

السؤال 3

```
JFrame frame = new JFrame();
frame.setVisible(true);
frame.add(label);
```

ما المشكلة؟

الإجابة:
يجب إضافة المكونات قبل:

```
frame.setVisible(true);
```

السؤال 4

```
ButtonGroup group = new ButtonGroup();

JRadioButton r1 = new JRadioButton("A");
JRadioButton r2 = new JRadioButton("B");
```

ما الخطأ؟

الإجابة:
لم تتم إضافة الأزرار إلى الـ ButtonGroup:

```
group.add(r1);
group.add(r2);
```

Lab 7 (Multithreading)

أكمل الكود

السؤال 5

```
Thread t = new Thread(() -> {
    while(seconds > 0){
        System.out.println(seconds);
        seconds--;

        try{
            ____________;
        }catch(Exception e){}
    }
});
```

الإجابة:

```
Thread.sleep(1000);
```

السؤال 6

```
synchronized(lock){
    while(paused){
        ____________;
    }
}
```

الإجابة:

```
lock.wait();
```

السؤال 7

```
synchronized(lock){
    paused = false;
    ____________;
}
```

الإجابة:

```
lock.notify();
```

أوجد الخطأ

السؤال 8

```
if(paused){
    lock.wait();
}
```

ما الخطأ؟

الإجابة:
يجب استدعاء wait داخل synchronized:

```
synchronized(lock){
    lock.wait();
}
```

السؤال 9

```
thread.sleep(1000);
```

ما الخطأ؟

الإجابة:
الصحيح:

```
Thread.sleep(1000);
```

السؤال 10

```
Thread timerThread;

resetButton.addActionListener(e -> {
```

```
    timerThread.stop();
});
```

ما المشكلة؟

الإجابة:
استخدام `stop()` غير آمن ومهمل Deprecated.

يفضل:

```
timerThread.interrupt();
```

Lab 8 (Lambda & Streams)

أكمل الكود

السؤال 11

```
List<String> names =
products.stream()
        .map(Product::getName)
        ____________
        .collect(Collectors.toList());
```

لترتيب الأسماء.

الإجابة:

```
.sorted()
```

السؤال 12

```
double avg =
products.stream()
        .collect(Collectors.__________(
            Product::getPrice
        ));
```

الإجابة:

```
averagingDouble
```

السؤال 13

```
Map<String,List<Product>> map =
products.stream()
```

```
        .collect(Collectors._____(
            Product::getCategory
        ));
```

الإجابة:

```
groupingBy
```

أوجد الخطأ

السؤال 14

```
products.stream()
        .filter(p -> {
            p.getPrice() > 100;
        })
        .collect(Collectors.toList());
```

ما الخطأ؟

الإجابة:
لا يوجد return.

الصحيح:

```
.filter(p -> p.getPrice() > 100)
```

أو

```
.filter(p -> {
    return p.getPrice() > 100;
})
```

السؤال 15

```
products.stream()
        .map(Product::getPrice)
        .average();
```

ما الخطأ؟

الإجابة:
average() تعمل على DoubleStream وليس Stream.

الصحيح:

```
products.stream()
        .mapToDouble(Product::getPrice)
```

```
        .average();
```

أسئلة نظرية متوقعة

ما الفرق بين sleep() و wait() ؟

sleep()

- توقف الـ Thread لمدة محددة.
- لا تحرر الـ Lock.

wait()

- تستخدم داخل synchronized.
- تحرر الـ Lock وتنتظر notify().

لماذا نستخدم ButtonGroup ؟

للسماح باختيار Radio Button واحد فقط.

لماذا نستخدم SwingUtilities.invokeLater() ؟

لتشغيل واجهة Swing على Event Dispatch Thread بطريقة آمنة.

ما فائدة Streams ؟

- تقليل الكود.
- معالجة البيانات بطريقة Functional.
- دعم عمليات map و filter و collect بسهولة.

Lab 6 (GUI Quiz)

السؤال 1

```
int score = 0;

score++;
score++;

System.out.println(score);
```

الناتج:

```
2
```

السؤال 2

```
int currentQuestionIndex = 1;

currentQuestionIndex++;

System.out.println(currentQuestionIndex);
```

الناتج:

```
2
```

السؤال 3

```
JButton btn = new JButton("Next Question");

System.out.println(btn.getText());
```

الناتج:

```
Next Question
```

السؤال 4

```
String selected = "Paris";

System.out.println("Your selection: " + selected);
```

الناتج:

```
Your selection: Paris
```

السؤال 5

```
int score = 0;

boolean correct = true;

if(correct)
    score++;

System.out.println(score);
```

الناتج:

```
1
```

Lab 7 (Multithreading)

السؤال 6

```
int seconds = 5;

while(seconds > 0){
    System.out.print(seconds + " ");
    seconds--;
}
```

الناتج:

```
5 4 3 2 1
```

السؤال 7

```
boolean paused = false;

if(paused)
    System.out.println("Paused");
else
    System.out.println("Running");
```

الناتج:

```
Running
```

السؤال 8

```
int seconds = 2;
```

```
while(seconds >= 0){
    System.out.println(seconds);
    seconds--;
}
```

الناتج:

```
2
1
0
```

السؤال 9

```
String text = "Time's up!";

System.out.println(text);
```

الناتج:

```
Time's up!
```

السؤال 10

```
boolean paused = true;

paused = !paused;

System.out.println(paused);
```

الناتج:

```
false
```

Lab 8 (Lambda & Streams)

السؤال 11

```
List<Integer> nums =
Arrays.asList(1,2,3);

nums.stream()
    .forEach(System.out::println);
```

الناتج:

```
1
```

```
2
3
```

السؤال 12

```
List<Integer> nums =
Arrays.asList(1,2,3,4,5);

nums.stream()
    .filter(n -> n % 2 == 0)
    .forEach(System.out::println);
```

الناتج:

```
2
4
```

السؤال 13

```
List<Integer> nums =
Arrays.asList(1,2,3);

nums.stream()
    .map(n -> n * 10)
    .forEach(System.out::println);
```

الناتج:

```
10
20
30
```

السؤال 14

```
List<String> names =
Arrays.asList("Ali","Sara");

names.stream()
    .map(String::toUpperCase)
    .forEach(System.out::println);
```

الناتج:

```
ALI
SARA
```

السؤال 15

```
List<Integer> nums =
Arrays.asList(10,20,30);

int sum =
nums.stream()
    .mapToInt(Integer::intValue)
    .sum();

System.out.println(sum);
```

الناتج:

```
60
```

السؤال 16

```
List<Integer> nums =
Arrays.asList(1,2,3,4);

long count =
nums.stream()
    .filter(n -> n > 2)
    .count();

System.out.println(count);
```

الناتج:

```
2
```

السؤال 17

```
List<String> items =
Arrays.asList("Laptop","Mouse","Monitor");

items.stream()
     .filter(s -> s.length() > 5)
     .forEach(System.out::println);
```

الناتج:

```
Laptop
Monitor
```

السؤال 18

```
List<Integer> nums =
Arrays.asList(1,2,3,4);
```

```
nums.stream()
    .map(n -> n * 2)
    .filter(n -> n > 4)
    .forEach(System.out::println);
```

الناتج:

```
6
8
```

السؤال 19

```
double avg =
Arrays.asList(10,20,30)
      .stream()
      .mapToInt(Integer::intValue)
      .average()
      .getAsDouble();

System.out.println(avg);
```

الناتج:

```
20.0
```

السؤال 20 (شامل وقد يأتي في الاختبار)

```
List<String> products =
Arrays.asList("Laptop","Mouse","Phone");

products.stream()
        .filter(p -> p.length() > 5)
        .map(String::toUpperCase)
        .forEach(System.out::println);
```

الناتج:

```
LAPTOP
PHONE
```

لأن:

- Laptop ✓
- Mouse ✗ (طوله 5)
- Phone ✓

ثم تتحول إلى أحرف كبيرة.

Lab 6 + OOP

السؤال 1

```
class Quiz {
    int score = 0;

    void answer(boolean correct){
        if(correct)
            score++;
    }
}

public class Main {
    public static void main(String[] args) {

        Quiz q = new Quiz();

        q.answer(true);
        q.answer(false);
        q.answer(true);

        System.out.println(q.score);
    }
}
```

الناتج

```
2
```

السؤال 2

```
String selected = null;

if(selected == null)
    selected = "A";

System.out.println(selected);
```

الناتج

```
A
```

السؤال 3

```
int currentQuestionIndex = 0;
```

```
for(int i=0;i<3;i++)
    currentQuestionIndex++;

System.out.println(currentQuestionIndex);
```

الناتج

```
3
```

Lab 7 (Multithreading)

السؤال 4

```
public class Main {
    public static void main(String[] args) {

        Thread t = new Thread(() -> {
            for(int i=1;i<=3;i++)
                System.out.print(i + " ");
        });

        t.start();

        System.out.print("A ");
    }
}
```

الناتج

100% غير محدد

قد يكون:

```
A 1 2 3
```

أو

```
1 2 3 A
```

لأن الـ Thread يعمل بالتوازي أو أي ترتيب مشابه.

السؤال 5

```
Object lock = new Object();
```

```
synchronized(lock){

    System.out.println("Start");

    synchronized(lock){
        System.out.println("Inside");
    }

    System.out.println("End");
}
```

الناتج

```
Start
Inside
End
```

السؤال 6

```
int seconds = 3;

while(seconds >= 0){

    if(seconds == 0){
        System.out.println("Time's up!");
    }else{
        System.out.println(seconds);
    }

    seconds--;
}
```

الناتج

```
3
2
1
Time's up!
```

السؤال 7

```
boolean paused = true;

if(paused){
    paused = false;
}

System.out.println(paused);
```

الناتج

```
false
```

Lab 8 (Streams)

السؤال 8

```
List<Integer> nums =
Arrays.asList(1,2,3,4,5);

nums.stream()
    .filter(n -> n % 2 != 0)
    .map(n -> n * n)
    .forEach(System.out::println);
```

الناتج

```
1
9
25
```

السؤال 9

```
List<String> names =
Arrays.asList("Ali","Sara","Ahmed");

names.stream()
     .filter(n -> n.length() > 4)
     .map(String::toUpperCase)
     .forEach(System.out::println);
```

الناتج

```
AHMED
```

السؤال 10

```
List<Integer> nums =
Arrays.asList(5,10,15);

int result =
nums.stream()
    .mapToInt(Integer::intValue)
    .sum();

System.out.println(result);
```

الناتج

السؤال 11

```
List<Integer> nums =
Arrays.asList(2,4,6);

double avg =
nums.stream()
    .mapToInt(Integer::intValue)
    .average()
    .orElse(0);

System.out.println(avg);
```

الناتج

```
4.0
```

السؤال 12

```
List<String> products =
Arrays.asList(
    "Laptop",
    "Mouse",
    "Monitor",
    "Phone"
);

products.stream()
        .filter(p -> p.length() > 5)
        .sorted()
        .forEach(System.out::println);
```

الناتج

```
Laptop
Monitor
```

السؤال 13 (صعب)

```
List<Integer> nums =
Arrays.asList(1,2,3,4);

nums.stream()
    .map(n -> n + 1)
    .filter(n -> n % 2 == 0)
    .forEach(System.out::println);
```

تتبع التنفيذ

```
1 → 2 → يطبع
2 → 3 → لا يطبع
3 → 4 → يطبع
4 → 5 → لا يطبع
```

الناتج

```
2
4
```

السؤال 14 (صعب جداً)

```
List<String> names =
Arrays.asList("Ali","Sara","Ahmed","Omar");

long count =
names.stream()
     .filter(n -> n.contains("a"))
     .count();

System.out.println(count);
```

الناتج

```
1
```

لأن Java حساسة لحالة الأحرف:

```
Ali   ❌
Sara  ✔
Ahmed ❌
Omar  ❌
```

السؤال 15 (من أكثر الأسئلة شيوعاً)

```
List<Integer> nums =
Arrays.asList(1,2,3,4,5);

nums.stream()
    .filter(n -> n > 2)
    .map(n -> n * 10)
    .forEach(System.out::println);
```

الناتج

```
30
40
```

سؤال دمج OOP + Streams (مستوى اختبار)

```
class Product {

    String name;
    double price;

    Product(String n,double p){
        name = n;
        price = p;
    }
}

public class Main {

    public static void main(String[] args) {

        List<Product> products = Arrays.asList(
                new Product("Laptop",1200),
                new Product("Mouse",25),
                new Product("Monitor",300)
        );

        products.stream()
                .filter(p -> p.price > 100)
                .map(p -> p.name)
                .forEach(System.out::println);
    }
}
```

الناتج

```
Laptop
Monitor
```

لأن فيه Lab 8 هذا النوع غالبًا يأتي في الاختبار النهائي للـ:

- Objects
- filter()
- map()
- Stream Pipeline كاملة
- تتبع الناتج بدقة.