

# JAVA PRÜFUNG II | VARIANTE C

## Bedingungen:

- Erlaubte Hilfsmittel: Unterrichtsunterlagen, Java Buch, Übungen und alte Prüfungen.
- Die Prüfung ist schrittweise, gemäss Aufgabenstellung lokal auf Ihrem Computer zu lösen. Kopieren sie zu diesem Zwecke den gesamten Ordner *P2\_HS16\_17\_SchirmSchoner\_Vorlage* auf Ihre lokale Harddisk und importieren sie das Projekt in Eclipse. Am Ende der Prüfung ist der Ordner *src* umbenannt in *NameVorname* abzugeben.
- Setzen sie als erstes Ihren Namen und Vornamen in die Dateien.
- Gegenseitiges Abschreiben in irgendeiner Form führt für alle Beteiligten zur Note 1!
- Folgend sie bei der Wahl von Variablen exakt den Angaben in der Aufgabenstellung.
- Die Beilage mit dem Layout muss unterschrieben zurückgegeben werden!

## Beschreibung:

Ziel dieser Prüfung ist es, einen Bildschirmschoner zu programmieren. Das Programm zeigt eine Aufnahme des berühmten Big-Ben in London mit laufender Uhr. Die Uhr verfügt über den berühmten Big-Ben Glockenschlag, der jede Viertelstunde entsprechend der Zeit erklingt. Im Weiteren kommt es zufällig zu gelegentlichen Gewittern. Dabei nimmt die Blitzwahrscheinlichkeit nach einem ersten Blitz zu, das Gewitter intensiviert sich und die Wahrscheinlichkeit nimmt anschliessend wieder ab. Die Applikation kann im Vollbild- oder im Teilbildmodus verwendet werden.



Die Applikation verfügt auf dem Kontroll-Panel im unteren Bereich über einige User-Interface-Elemente, die nach rund 10 Sekunden inaktiver Maus verschwinden und auf eine Bewegung hin wieder erscheinen:

- Der Exit - Button erlaubt es, die Applikation zu schliessen.

- Der Test - Button und die beiden Eingabefelder dienen Testzwecken: Mit ihnen kann der Glockenschlag für die definierte Zeit getestet werden.

Das Programm ist in der alt bekannten Art, wie im Klassendiagramm gezeigt, aufgebaut: Die *Schirm-SchonerApplikation* ist ein *JFrame* und hat die *TopView*. Die *TopView* ist ein *JPanel* und beheimatet das *KontrollPanel* sowie die *Uhr*. Sie verfügt über einen *MouseListener* um auf Mausbewegungen reagieren zu können und hat den *Timer*, der alle Sekunde via *SimpleTimerListener* die zugehörige Methode *timerAction()* auslöst. Das *KontrollPanel* besitzt die beiden *JButton* und die *JTextField*. Die *JButton* sind mit dem zugehörigen *ActionListener* verbunden und lösen die Methode *actionPerformed()* aus. Das *KontrollPanel* hat weiter die Referenz auf DIE *Uhr*, um die Methode *schlag()* auslösen zu können. Die *Uhr* selbst ist ein *GrafikObjekt* und hat ein *HGBildObjekt* sowie die drei Zeiger, die je von der Superklasse *Zeiger* erben, die wiederum ein *GrafikObjekt* ist. Das *HGBildObjekt* ist ein *GrafikObjekt* und hat ein Hintergrundbild sowie die drei Blitzbilder, die in einem Array abgelegt sind. Zu jedem Blitzbild gehört ein zugehöriger *Audioclip* mit dem zugehörigen Donner. Die Uhr hat weiter drei *AudioClips* mit den 15/30/45 Minutenschlägen. Die zwölf Stundenschläge sind in einem Array abgelegt.

Die Aufgabenstellung führt Sie schrittweise zum Ziel. Folgen Sie daher beim Lösen der Prüfung der Aufgabenstellung. Sämtliche Bilder sind im Ordner *src/bilder* zu finden und selbstsprechend bezeichnet.

**Achten Sie darauf, dass Ihr Code kompilierbar bleibt!**

**Das blosse Einsammeln von *billigen Punkten* wird nicht honoriert!**

### Aufgabe 1: Erstellen des Grundgerüsts (~4 Pte.)

Als erstes wollen wir das Grundgerüst des Programms anhand des Klassendiagramms kontrollieren und allenfalls ergänzen. Viele Elemente der Klassen sind dabei bereits gegeben.

- a) Ergänzen Sie sämtliche *Attribut-* und *Methodendeklarationen* gemäss Klassendiagramm. Machen Sie sich dabei mit den vorhandenen Attributen und Methoden-Deklarationen bekannt.

### Aufgabe 2: Klasse *TopView* (~21 Pte.)

Die *TopView* ist ein *JPanel* und beheimatet das *KontrollPanel*, die *Uhr*, einen *MouseMotionListener* und den *Timer*.

- a) Kompletieren Sie die Klasse *TopView* gemäss Dokumentation und Layout.

### Aufgabe 3: Klasse *KontrollPanel* (~13 Pte.)

Das *KontrollPanel* besitzt die beiden *JButton*, die *JTextField* und implementiert den *ActionListener*. Das *KontrollPanel* hat weiter die Referenz auf DIE *Uhr*.

- a) Kompletieren Sie die Klasse *KontrollPanel* gemäss Dokumentation und Layout

### Aufgabe 4: Klasse *GrafikObjekt* (~2 Pte.)

Alle Objekte der Uhr basieren auf *GrafikObjekt*. Die Klasse beinhaltet die x,y-Position der Objekte sowie die für alle *GrafikObjekte* benötigte Skalierung in X- und Y-Richtung, damit die Applikation bei beliebiger Bildschirmgrösse korrekt erscheint.

- a) Implementieren Sie die Methoden der Klassen gemäss Dokumentation im Code.

### Aufgabe 5: Klassen *Zeiger*, *SekundenZeiger*, *MinutenZeiger*, *StundenZeiger* (~11 Pte.)

Die genannten vier Klassen sind alle in der Datei *Zeiger.java* zu finden. Basierend auf der Klasse *Zeiger* werden der benötigte Sekunden-, Minuten- und Stundenzeiger implementiert.

- b) Implementieren Sie die Methoden der Klassen gemäss Dokumentation im Code.

### Aufgabe 6: Klasse *HGBildObjekt: Teil I* (~2 Pte.)

Das *HGBildObjekt* ist ein *GrafikObjekt*, hat ein Hintergrundbild sowie die drei Blitzbilder, die in einem Array abgelegt sind. Zu jedem Blitzbild gehört ein zugehöriger *Audioclip* mit dem zugehörigen Donner.

- a) Implementieren Sie die Methoden der Klasse *HGBildObjekt* gemäss Dokumentation in der Klasse aber noch **OHNE Teil II** (Gewitter).

### Aufgabe 7: Klasse *Uhr* (~28 Pte.)

Die *Uhr* selbst ist ein *GrafikObjekt* und hat ein *HGBildObjekt* sowie die drei Zeiger und die diversen *AudioClips*.

- a) Implementieren Sie die Methoden der Klasse *Uhr* gemäss Dokumentation in der Klasse.

### Aufgabe 8: Klasse *HGBildObjekt: Teil II* (~16 Pte.)

- a) Kompletieren Sie die Methoden der Klasse *HGBildObjekt* gemäss Dokumentation in der Klasse **inklusive Teil II** (Gewitter).
- b) Ergänzen Sie den Code um noch allfällig fehlenden Code.
- c) Sollte die Applikation nun funktionieren, so schlagen Sie über die Lautsprecher 12:00 und lassen Sie es einmal so richtig Donnern ...