|  |
| --- |
| Studieneinheit / Modul: Coding - Projektarbeit (Modulabschlussprüfung) |
| Studiengang: Elektro- und Informationstechnik (B. Eng.) (1648.01) |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Prüfung: | Coding - Projektarbeit (Modulabschlussprüfung) (CODI PAxx) | | Abgabedatum: | 14.07.2025\* | | Name: | **Tim Gassan** | | Matrikelnummer: | **910071405** | |

**Bitte berücksichtige die folgenden allgemeinen Hinweise!**

**Vorbemerkung:**

Die Projektarbeit ist bis zu dem oben genannten Termin einzureichen. Verspätete oder falsche eingereichte Lösungen können nicht gewertet werden.

Sende die vollständige Aufgabenstellung und Lösung über den *Online-Campus an der gleichen Stelle ein, an der du die Aufgabe angefordert haben.* **Die Projektarbeit muss zwingend im Format Microsoft Word (.docx) erstellt und eingereicht werden.** Du darfst bei der Projektarbeit aber auch eine **zusätzliche** PDF-Datei einreichen.

Sollten in der Aufgabenstellung weitere Dateien gefordert werden, darfst du das in der Aufgabenstellung genannte Format zusätzlich einreichen. Grundsätzlich sind aber nur die Formate PDF (.pdf), Microsoft Excel (.xls/.xlsx), MP3 (.mp3) oder Powerpoint (.ppt/.pptx) möglich. Bis maximal drei Dateien sind möglich.

Sollte es Abweichungen der Einreichform geben, findest du diese Hinweise in der Aufgabenstellung.

Bitte berücksichtige, dass bei Einreichung der Lösung zunächst alle Dateien ausgewählt werden müssen, bevor der Button „Lösung versenden“ betätigt wird.

Die Lösung darf nur eine maximale Größe von 8 MB haben.

Sollte es eine Vorgabe zum Umfang der Wortanzahl in Ihrer Aufgabenstellung geben, gilt für dich folgendes: Wortzahlüberschreitungen sind maximal in Höhe von 10% der vorgesehenen Wortzahl möglich. Wird die vorgegebene Wortzahlbegrenzung zuzüglich 10% Toleranzgrenze überschritten, führt dies zu einem pauschalen Punktabzug in Höhe der Prozentzahl der Wortzahlüberschreitung.

Bitte beachte die „Checkliste zur Selbsteinschätzung eines Forschungsprojekts auf ethische Unbedenklichkeit“. Sofern es sich bei der vorliegenden Arbeit um ein empirisches Forschungsvorhaben mit menschlichen Probandinnen und Probanden handelt, musst du die in der Checkliste genannten Anforderungen beachten. Die weiteren Hinweise hierzu findest du im Download Bereich auf dem Online Campus.

**Hinweis zur Bewertung:**

Die Prüfungs- oder Studienleistung ist bestanden, wenn sie mindestens mit ausreichend (4,0) oder besser bewertet wurde.

**Hinweise zur Bearbeitung:**

Solltest du für deine Lösung externe Quellen hinzuziehen oder fremdes Gedankengut verwenden, so ist dieses kenntlich zu machen. Die besonderen Hinweise zu dieser Projektarbeit findest du ggf. über der Aufgabenstellung

Bei Literaturempfehlungen kann nicht durchgängig gewährleistet werden, dass diese jederzeit kostenfrei über die von uns angebotenen Bibliotheken verfügbar sind. Im Regelfall kannst du die Literatur auch an anderen Stellen, etwa Online-Bibliotheken, beziehen. Grundsätzlich bist du frei in der Auswahl der für die Bearbeitung der Hausarbeit geeigneten Literatur, sodass du auch in den bereit gestellten Online-Bibliotheken alternative und ggf. kostenfreie Literatur heraussuchen können.

**Viel Erfolg!**

**Aufgabe:**

Erstellen Sie mit Java und Swing ein vollständig im Sinne der folgenden Aufgabenstellung lauffähiges Grafikprogramm mit grafischer Oberfläche. Das Programm soll über eine Menü- und Symbolleiste zur Auswahl von Funktionen und einen Bereich für das Erstellen von Grafik verfügen.

Es soll mindestens folgende Funktionen unterstützen:

* Zeichnen von Ellipsen, Linien und Rechtecken mit der Maus,
* Erstellen neuer Dateien,
* Laden von Dateien und
* Speichern von Grafiken.

Ergänzen Sie außerdem noch mindestens drei weitere Funktionen Ihrer Wahl – zum Beispiel eine Farbauswahl für die grafischen Elemente oder grafische Werkzeuge wie ein Radiergummi.

Stellen Sie den Nutzenden für das Laden und Speichern von Dateien entsprechende Dialoge zur Verfügung. Sie sollen nur Dateien vom Typ JPG anzeigen. Für die Dialoge können Sie die Klasse JFileChooser verwenden. Filter für die Dialoge erstellen Sie über die Klasse FileFilter.

Für den Bereich zum Erstellen von Grafiken können Sie eine Komponente vom Typ BufferedImage verwenden. Für die Zeichenoperationen beschaffen Sie sich den Grafikkontext vom Typ Graphics2D für die Komponente vom Typ BufferedImage. Zum Laden und Speichern von Grafiken können Sie die Klasse ImageIO verwenden. Zum Verarbeiten von Mausereignissen verwenden Sie die Klasse MouseAdapter.

**Bitte beachten Sie bei der Umsetzung folgende Punkte:**

Stellen Sie den Nutzenden Tastaturkommandos für das Erstellen neuer Dateien sowie für das Laden und Speichern von Dateien zur Verfügung. Auch die Bedienung des Menüs soll über die Tastatur möglich sein. Benutzen Sie dabei übliche Kürzel beziehungsweise Tastenkombinationen.

Verwenden Sie für die Symbolleiste übliche Symbole und versehen Sie die Symbole mit Tooltipps. Eine Sammlung mit Symbolen finden Sie zum Beispiel im Internet auf der Seite https://www.oracle.com/java/technologies/java-look-and-feel-graphics-repository.html.

Vor dem Anlegen einer neuen Datei soll eine Abfrage erscheinen, ob wirklich eine neue Datei angelegt werden soll.

Achten Sie auf die Betriebssicherheit des Programms – zum Beispiel beim Laden und Speichern von Dateien.

Beschreiben Sie ausführlich und fachlich Ihr Vorgehen beim Erstellen des Programms. Kommentieren Sie umfassend Ihren Code. Beachten Sie bei der Umsetzung Vorgaben wie Clean Code und führen Sie systematische Tests Ihres Programms durch.

**Reichen Sie 2 Dateien als Worddatei ein:**

1. Projektdokumentation (Vorgehensbeschreibung, Qualitätssicherungs- und Testdokumentation)
2. Vollständiger, kommentierter Java-Code ohne jegliche zusätzlichen Texte oder Elemente, der reine Code.

**Projektdokumentation:**

1. **Vorgehensbeschreibung**

Das Abschlussprojekt ist ein Zeichenprogramm nach obiger Aufgabenstellung. Um dieses realisieren zu können bestand der erste Schritt darin, sich einen groben Überblick über die bis dahin unbekannten Klassen für grafische Tools sowie das Laden und Speichern von Bildern zu verschaffen. Im gleichen Zeitrahmen wurde ein Gradle-Projekt in IntelliJ und ein zugehöriges Git-Repository erstellt.

Anschließend ging es an die Planung des Projekts um die Frage zu beantworten, welche Funktionen das Programm im finalen Zustand haben soll. Die detaillierte Auflistung ist in der Qualitätssicherungsdokumentation zu finden. Es wurde zu großen Teilen mit Klassen gearbeitet werden, von denen die möglichen Funktionen erst im Laufe des Programmierens entdeckt wurden. Daher ist im Laufe des Projekts immer wieder zu diesem Schritt zurückgekehrt worden, um das Programmziel an das neue Wissen anzupassen. Dabei wurde darauf geachtet, dass der Umfang des Projekts nicht mit immer neuen Funktionen zu groß wurde, sondern im ursprünglich geplanten Rahmen bleibt.

Im nächsten Schritt wurde die Struktur des Programms festgelegt. Dabei wurde sich für eine Aufteilung in drei Klassen entschieden. Die Klasse „PaintProgram“, welche die Main-Funktion enthält, die Klasse „Frame“, die das Fenster des Programms abbildet und die Klasse „PaintPanel“, die die Zeichenfläche darstellt. Zu diesem Zeitpunkt wurde auch festgelegt, dass das Programm in Englisch gecodet wird, mit der Ausnahme von sichtbarem Text im Programm – es ist für einen deutschen Anwender programmiert – und Kommentaren.

Der nachfolgende Teil hat sich mit dem eigentlichen Programmieren beschäftigt. An dieser Stelle soll eine kurze Zusammenfassung der implementierten Funktionen folgen, konkrete Hinweise zum Code stehen in den Kommentaren. Es wurde das JFrame und das darauf abzubildende JPanel erstellt und die Menü- und die Symbolleiste zunächst mit Platzhaltern erstellt. Anschließend wurden die großen Funktionen wie die Zeichenwerkzeuge, die Farbauswahl und die Strichdickenauswahl hinzugefügt. Es wurden dann die Funktionen zum Erstellen neuer Zeichenflächen erstellt, die mit Sicherheitsabfrage entweder direkt ein leeres Blatt in gleicher Größe oder über Benutzerabfrage ein leeres Blatt mit neuer Größe erstellen. Es kamen die Funktionen zum Speichern- und Laden der Dateien hinzu, die standardmäßig im Projektordner zu finden sind. Die Funktionen haben ihre Hotkeys und Tooltips bekommen und es wurden Scrollbars bei zu großen Zeichenflächen hinzugefügt.

Im nächsten großen Schritt wurden die Feinheiten bearbeitet. Dazu gehört zum Beispiel das Layout der Symbolbar sowie die Icons der Buttons, die bis dahin noch Platzhalter waren. Das Markieren der Buttons, wenn sie angewählt sind ist dazugekommen und als Abschluss eine eigentlich große Funktion, die bis zum Ende nicht sicher in der finalen Version enthalten war, weil nicht sicher war, wie aufwendig sie sein würde. Letztendlich hat sich die Implementation der Vorschaubilder beim Ziehen von Linien, Ellipsen und Rechtecken als machbar erwiesen. Währen dieses gesamten Prozesses wurde darauf geachtet, nach möglichst kleinen, vollständigen Änderungen einen neuen Commit und Push auf GitHub auszuführen, um bei Problemen möglichst sicher und flexibel zu sein.

Im letzten großen Schritt stand das Testen des Programms auf dem Plan. Zunächst wurde sich bewusst dazu entschieden, auf das Schreiben von automatisierten Tests während der Programmierung zu verzichten, da durch neue Erkenntnisse über die grafischen Klassen Programmteile immer wieder Gefahr liefen, überarbeitet werden zu müssen. Das hätte bei automatisierten Tests zu Mehrarbeit geführt und gleichzeitig ist dieses Projekt vom Umfang her vergleichsweise klein, so dass das Testen ans Ende verschoben werden konnte. Im zweiten Punkt gibt es, neben Gettern und Settern, nur eine Methode im Projekt, die mit den bekannten JUnit-Tests automatisiert getestet werden kann. Für Variablen, für die ein Getter und ein Setter existiert, wurde ein Test für beide zusammen geschrieben. Die einzelnen Setter wurden nicht mit JUnit getestet und werden in den grafischen Tests mit überprüft. Für das automatisierte Testen des grafischen Teils dieses Programms wären extra Werkzeuge wie z. B. QF-Test notwendig. Auch an dieser Stelle wurde aus Zeitgründen darauf verzichtet. Stattdessen wurden die weiter unten dokumentierten Testfälle manuell durchgeführt. Für ein größeres Projekt sei angemerkt, dass dieses Vorgehen nicht zu empfehlen ist. Wenn das Programm öffentlich gemacht werden sollte, würden sich an dieser Stelle weitere Anwendungstests mit mehreren Testpersonen empfehlen, um Feedback zu erhalten und das Programm abzurunden.

Die beim Testen gefundenen Fehler wurden behoben und bei Bedarf, neue Testfälle hinzugefügt, um ein möglichst vollständiges Bild vom Zustand des Programms zu enthalten.

Der Abschluss bestand darin, diese Dokumentation zu verfassen und die Qualitätssicherungs- und Testdokumentation fertigzustellen.

1. **Qualitätssicherungsdokumentation**

Um die Qualität dieses Programms sicherzustellen sollen an dieser Stelle auf folgende Punkte eingegangen werden: die Entwicklungsrichtlinien, die Qualitätskriterien, die Versionskontrolle und das Testen.

2.1 Entwicklungsrichtlinien

In diesem Projekt sind folgende Entwicklungsrichtlinien zu Beginn definiert worden, die dann im Laufe des Projekts eingehalten werden sollten:

* Variablennamen werden beschreibend in Englisch gewählt.
* Es werden beschreibende Konstanten für Strings (z. B. Actioncommands) verwendet, wenn diese mehr als zwei- bis dreimal verwendet werden, um Tippfehler zu vermeiden.
* Das Programm soll so geschrieben werden, dass es möglichst einfach wartbar und erweiterbar ist. Dazu gehört, dass sich wiederholende Abläufe, wenn möglich in einer Methode zusammengefasst werden, und das Codeteile, die zu den gleichen Objekten gehören (z. B. das Setzen verschiedener Eigenschaften eines Buttons) auch örtlich beieinanderstehen.
* Das Programm wird detailliert auf Deutsch auskommentiert
  1. Qualitätskriterien

Hier sollen die konkreten Anforderungen an das Zeichenprogramm dargelegt werden, um sowohl dem Programmierer ein klares Ziel zu geben, sowie das fertige Produkt daran messen zu können. An dieser Stelle kann davon ausgegangen werden, dass diese Kriterien erfüllt wurden, wenn keine Anmerkungen in Klammern hinter diesen stehen.

* Die Oberfläche besteht aus einer Menü- und Symbolleiste sowie einer grafischen Fläche zum Zeichnen.
* Die Menüleiste enthält übliche Reiter und kann mit der Tastatur bedient werden
* In der Menüleiste sind die Menüs Datei, ~~Bearbeiten, Ansicht~~ und Werkzeuge zu finden. (Das Menü „Bearbeiten“ war für die Funktion „Rückgängig machen“ gedacht und das Menü „Ansicht“ war ursprünglich für das Anpassen der Zeichenfläche gedacht und wird nicht mehr benötigt)
* Im Menü Werkzeuge sind die Zeichentools zusätzlich zur Symbolleiste zu finden.
* Im Menü Datei kann ein neues Zeichenblatt erstellt, ein Bild geladen, das aktuelle Bild gespeichert und das Programm beendet werden. Bis auf das Beenden des Programms sind diese Funktionen mit Hotkey ausführbar.
* Beim Erstellen eines neuen Zeichenblattes, beim Laden einer Datei und beim Beenden soll der Benutzer gefragt werden, ob er diese Aktion wirklich durchführen möchte.
* Beim Erstellen eines neuen Zeichenblattes soll der Benutzer die Option haben, die Größe des Zeichenblattes zu ändern
* ~~Im Menü Bearbeiten soll die Funktion „Rückgängig machen“ sein, mit der die letzte Aktion auf der Zeichenfläche zurückgenommen werden kann.~~ (Funktion gestrichen, da die Implementierung zu aufwändig gewesen wäre.)
* Die Symbolleiste enthält die Werkzeuge zum Zeichnen, die mit Icons und Tooltips zeigen, wofür sie gedacht sind.
* Die Werkzeuge sind über nachvollziehbare Hotkeys auswählbar.
* Die Zeichenwerkzeuge sind: ein Pinsel zum freien Zeichnen, Werkzeuge zum Zeichnen von Linien, Ellipsen und Rechtecken und ein Radierer zum freien Radieren. Die Werkzeuge für die Figuren sollen dabei beim Ziehen mit gedrückter Maustaste eine Vorschau anzeigen und in alle vier Richtungen gezogen werden können.
* Zum Arbeiten auf der Zeichenfläche soll die linke Maustaste verwendet werden.
* Wenn beim Ziehen mit einem der Figurenwerkzeuge die Zeichenfläche verlassen wird, soll die Figur sich trotzdem entsprechend auf der Zeichenfläche weiterverändern und gezeichnet werden, wenn die Maustaste losgelassen wird.
* Für alle Werkzeuge soll die Strichdicke mittels Textfeld und die Farbe über entsprechende Buttons eingestellt werden können. Diese Einstellung soll beim Wechsel des Werkzeugs erhalten bleiben. Ausnahme davon ist der Radierer. Bei ihm soll nichts passieren, wenn eine Farbbutton gedrückt wird. Die Größe des Radierers bleibt einstellbar.
* Die Buttons sollen dem Benutzer anzeigen, welches Werkzeug bzw. welche Farbe ausgewählt ist.
  1. Versionskontrolle

Während des Programmierens wurde regelmäßig das Repository aktualisiert und auch auf GitHub gepusht. Es wurde darauf geachtet, dass alle benötigten Dateien für das Programm im Repository sind, alle nicht benötigten Dateien z. B. der Ordner „savedPictures“ als Standardordner für das Laden und Speichern von Bildern aber nicht enthalten sind.

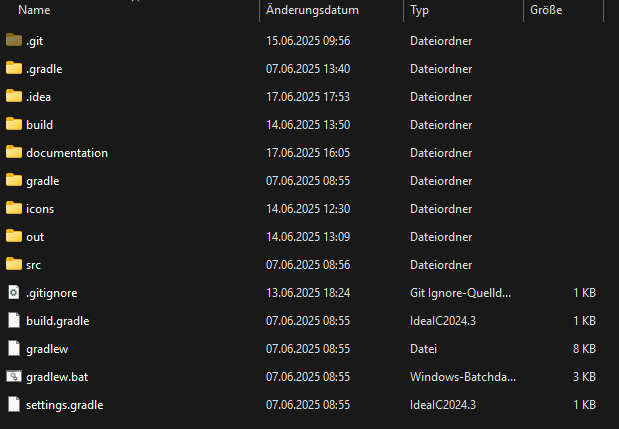


Abbildung Projektordner mit allen Dateien und Ordnern

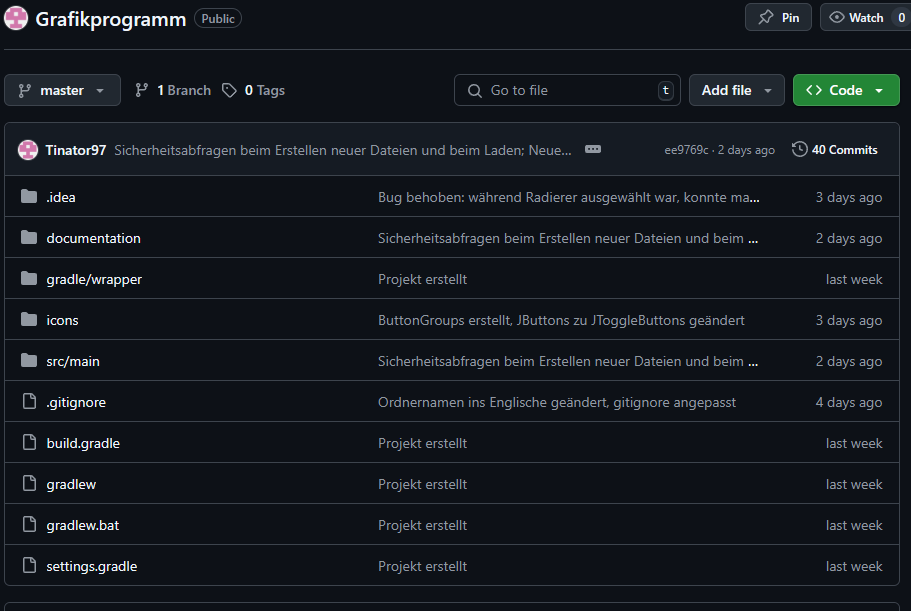


Abbildung Auf GitHub hochgeladene Dateien und Ordner

Im Folgenden ist die ist die Push-Historie vom Beginn des Projekts bis zur Fertigstellung zu sehen.

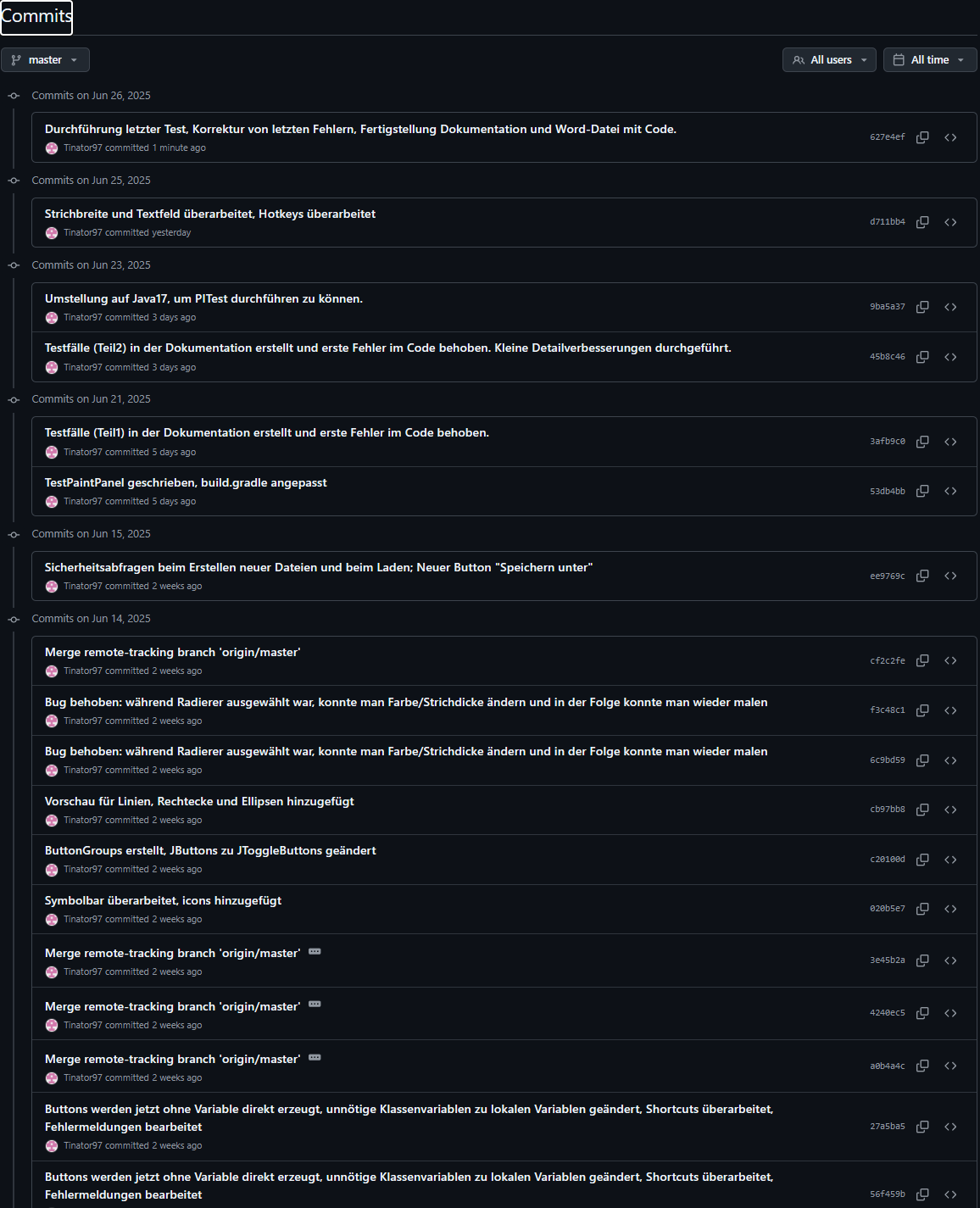


Abbildung Push-Verlauf auf GitHub 1 von 3

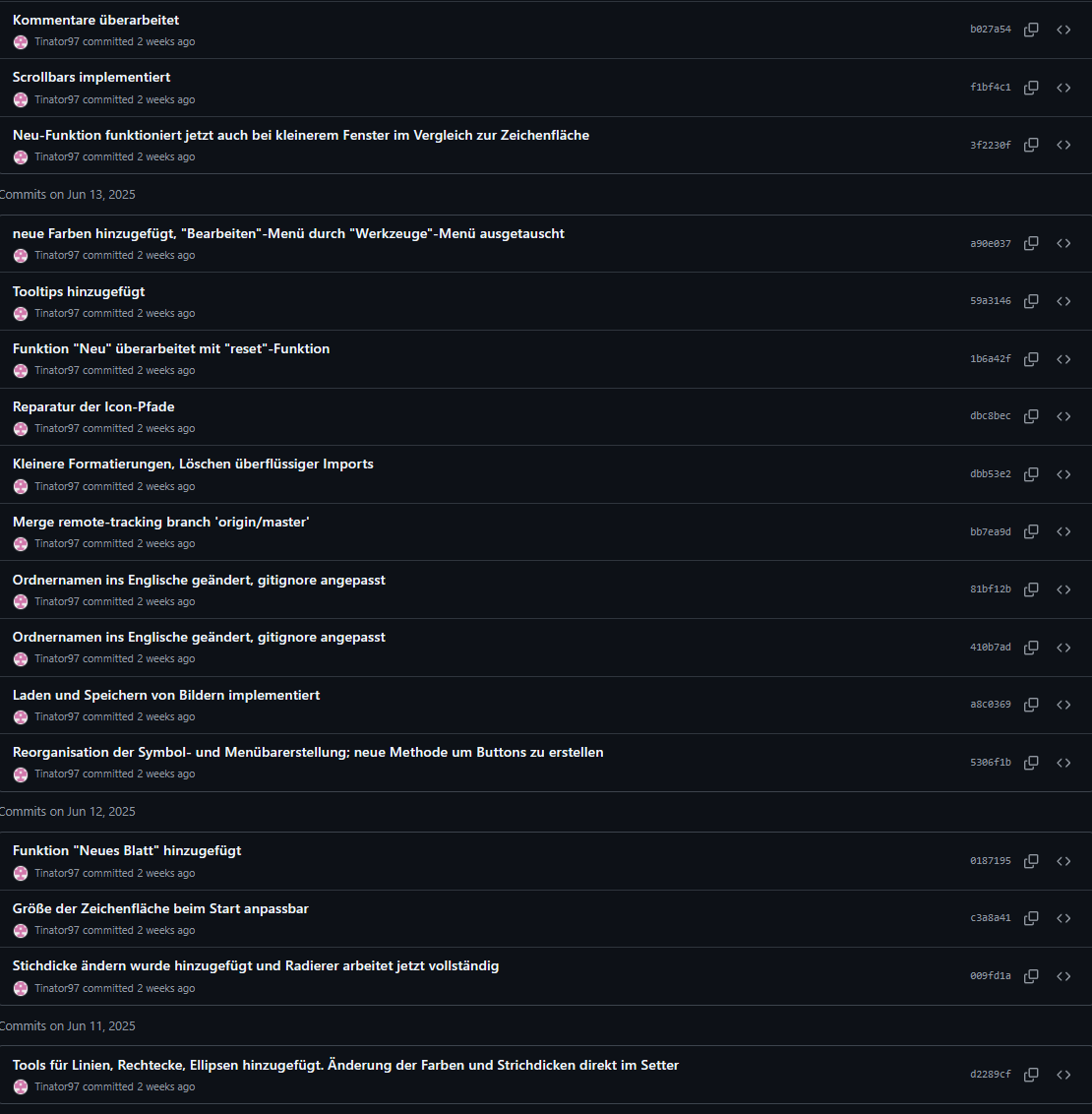


Abbildung Push-Verlauf auf GitHub 2 von 3

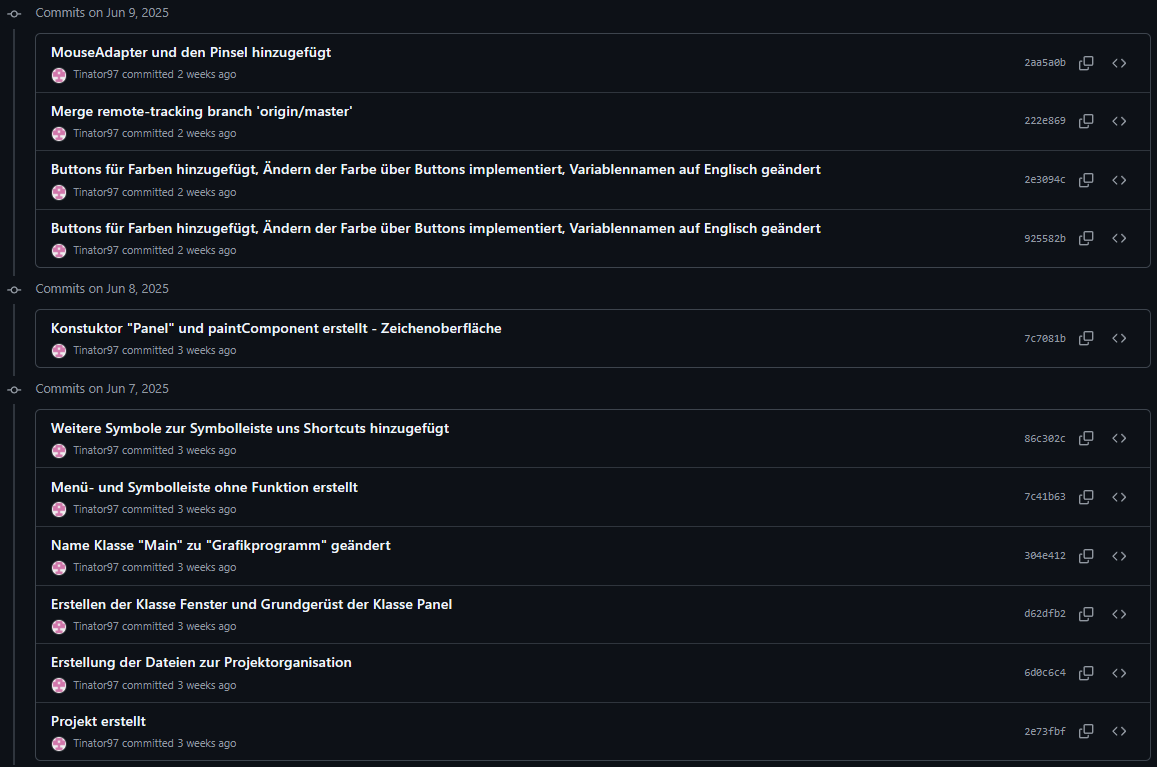


Abbildung Push-Verlauf auf GitHub 3 von 3

* 1. Testen

Das Testen nimmt einen wichtigen Platz ein, um sicherzustellen, dass das Programm allen Anforderungen genügt. Wie oben bereits beschrieben, wurde das Testen in diesem Projekt vornehmlich manuell und erst nach Abschluss des eigentlichen Programmierens vorgenommen.

Die Methode PaintPanel.checkOrientation ist die einzige Methode – abgesehen von Gettern und Settern– die mit bekannten JUnit-Tests automatisiert getestet werden kann. Das ist im entsprechenden Ordner „src/test“ zu finden. Für alle anderen Methoden ist grafisches Testen nötig, das mit speziellen Programmen auch automatisiert möglich wäre, in diesem Projekt aber manuell erfolgt.

Für alle Tests wurden entsprechende Grenzwerte und besondere Testfälle für Blackbox-Tests ausgewählt und weitere Testfälle für Whitebox-Tests hinzugefügt. Damit sollte jede Methode vollumfänglich getestet sein. Die genauen Tests mit ihren konkreten Testfällen sind im nächsten Abschnitt dokumentiert. Zusätzlich dazu wurde ein PITest durchgeführt. Dessen Aussagekraft ist allerdings nur marginal, da nur ein kleiner Teil mit JUnit getestet wurde.

1. **Testdokumentation**

Zunächst sollen die JUnit-Tests dokumentiert werden. Bei der Bezeichnung der Methoden wurde aufgrund der Übersichtlichkeit auf die Parameter verzichtet.

Genutzte Abkürzungen:

* GB: Grenzwert nach Blackbox-Verfahren
* ZB: Zusätzliche Werte nach Blackbox-Verfahren
* C2: Weitere Fälle für das Whitbox-C2-Verfahren
* NT: Test nicht durchgeführt (Begründung folgt)
* B: Bestanden
* NB: Nicht Bestanden

Test 1: Klasse: PaintPanel Methode: checkOrientation()

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Testfall**  **(Startpunkt, Endpunkt)** | **Bemerkung** | **Ergebnis** |
| **1** | (100/100), (500/500) | GB, Orientierung von links-oben nach rechts-unten | B |
| 2 | (500/500), (100/100) | GB, Orientierung von rechts-unten nach links-oben | B |
| 3 | (100/500), (500/100) | GB, Orientierung von links-unten nach rechts-oben | B |
| 4 | (500/100), (100/500) | GB, Orientierung von rechts-oben nach links-unten | B |
| 5 | (0/0), (100/100) | GB, vom Nullpunkt | B |
| 6 | (100/100), (0/0) | GB, nach Nullpunkt | B |
| 7 | (-100/-100), (-50,-50) | NT, Negative Koordinaten können auf dem Zeichenpanel nicht auftreten | - |
| 8 | (100/100), (100/500) | GB, vertikale Linie | B |
| 9 | (100/100), (500/100) | GB, horizontale Linie | B |
| 10 | (500/500), (500/500) | GB, Start- und Endpunkt der Gleiche | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 2: Klasse: PaintPanel Methode: setTool & getTool

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Testfall**  **(Tool)** | **Bemerkung** | **Ergebnis** |
| 1 | \*leer\* | ZB, erstes Beispiel | B |
| 2 | Brush | ZB, zweites Beispiel | B |
| 3 | Line | ZB, drittes Beispiel | B |
| 4 | Rectangle | ZB, viertes Beispiel | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 3: Klasse: PaintPanel Methode: setColor & getColor

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Testfall**  **(Color)** | **Bemerkung** | **Ergebnis** |
| 1 | Black | ZB, erstes Beispiel | B |
| 2 | Yellow | ZB, zweites Beispiel | B |
| 3 | Red | ZB, drittes Beispiel | B |
| 4 | Blue | ZB, viertes Beispiel | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 4: Klasse: PaintPanel Methode: setLastColor & getLastColor

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Testfall**  **(Color)** | **Bemerkung** | **Ergebnis** |
| 1 | Black | ZB, erstes Beispiel | B |
| 2 | Yellow | ZB, zweites Beispiel | B |
| 3 | Red | ZB, drittes Beispiel | B |
| 4 | Blue | ZB, viertes Beispiel | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 5: Klasse: PaintPanel Methode: setStroke & getStroke

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Testfall**  **(Strichdicke)** | **Bemerkung** | **Ergebnis** |
| 1 | \*leer\* | GB, leere Eingaben werden vorher durch Überprüfungen abgefangen und können nicht in dieser Funktion auftreten. Deshalb schlägt dieser Test fehl und ist im Programm auskommentiert. | NB |
| 2 | -5 | GB, negative Eingaben werden vorher durch Überprüfungen abgefangen und können nicht in dieser Funktion auftreten. Deshalb schlägt dieser Test fehl und ist im Programm auskommentiert. | NB |
| 3 | 0 | GB, die 0 wird vorher durch Überprüfungen abgefangen und können nicht in dieser Funktion auftreten. Deshalb schlägt dieser Test fehl und ist im Programm auskommentiert. | NB |
| 4 | 0.1 | GB | B |
| 5 | 1 | ZB, die erste ganze Zahl | B |
| 6 | 2 | ZB, eine zweite ganz Zahl | B |
| 7 | 1000 | ZB, eine große Zahl | B |
|  | 5.5 | ZB, eine weitere Gleitpunktzahl | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

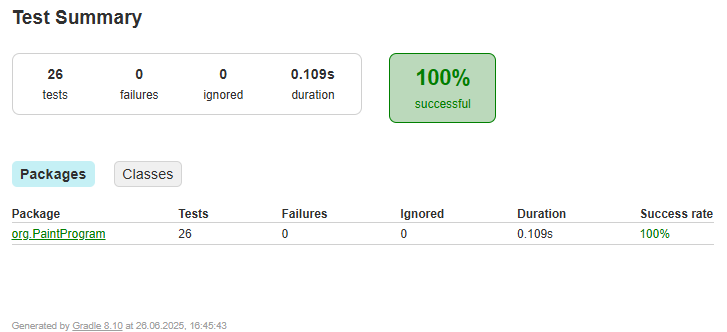


Abbildung Ergebnis der JUnit Tests

Als nächstes sollen die manuellen Tests für die grafischen Elemente dokumentiert werden. Es werden die gleichen Abkürzungen wie in den vorherigen Tests genutzt. Die Setter setStroke, setLastMousePosition, setEndPointOfShape warden automatisch innerhalb der folgenden Tests mitgetestet und werden deshalb nicht extra erwähnt. Da die grafischen Klassen teilweise voneinander Abhängig sind, können sich Testfälle für verschiedene Tests doppeln.

Test 6: Klasse: PaintPanel Methode: PaintPanel() (Konstruktor)

Voraussetzungen: -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bemerkung** | **Ergebnis** |
| 1 | Starten der Anwendung | Es erscheint ein weißes Panel mit den Abmaßen 1600 x 900. | ZB | B |
| 2 | Klicken und Ziehen auf dem Panel | Beim Klicken (und Ziehen) auf das Panel entsteht ein schwarzer Pinselstrich mit Strichdicke 5. | ZB | B |

Bemerkung: Da der Konstruktor nur beim Starten der Anwendung aufgerufen wird, bleibt es bei diesen Testfällen. Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 7: Klasse: PaintPanel Methode: paintComponent()

Voraussetzungen: -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bem.** | **Ergebnis** |
| 1 | Starten der Anwendung | Es erscheint ein weißes Panel, auf dem gezeichnet werden kann. | GB | B |
| 2 | Auswahl Linienwerkzeug, klicken und ziehen nach rechts unten | Es erscheint eine Vorschau der Linie vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 3 | Auswahl Linienwerkzeug, klicken und ziehen nach rechts oben | Es erscheint eine Vorschau der Linie vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 4 | Auswahl Linienwerkzeug, klicken und ziehen nach links oben | Es erscheint eine Vorschau der Linie vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 5 | Auswahl Linienwerkzeug, klicken und ziehen nach links unten | Es erscheint eine Vorschau der Linie vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 6 | Auswahl Rechteckwerkzeug, klicken und ziehen nach rechts unten | Es erscheint eine Vorschau des Rechtecks vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 7 | Auswahl Rechteckwerkzeug, klicken und ziehen nach rechts oben | Es erscheint eine Vorschau des Rechtecks vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 8 | Auswahl Rechteckwerkzeug, klicken und ziehen nach links oben | Es erscheint eine Vorschau des Rechtecks vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 9 | Auswahl Rechteckwerkzeug, klicken und ziehen nach links unten | Es erscheint eine Vorschau des Rechtecks vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 10 | Auswahl Ellipsenwerkzeug, klicken und ziehen nach rechts unten | Es erscheint eine Vorschau der Ellipse vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 11 | Auswahl Ellipsenwerkzeug, klicken und ziehen nach rechts unten | Es erscheint eine Vorschau der Ellipse vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 12 | Auswahl Ellipsenwerkzeug, klicken und ziehen nach rechts unten | Es erscheint eine Vorschau der Ellipse vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 13 | Auswahl Ellipsenwerkzeug, klicken und ziehen nach rechts unten | Es erscheint eine Vorschau der Ellipse vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 14 | Verlassen des Panels beim Ziehen | Die Vorschau verändert sich weiter mit dem Mauszeiger | GB | B |
| 15 | Auswahl Linienwerkzeug und Klicken auf Fläche, nachdem andere Linie gezeichnet wurde. | Es entsteht keine Vorschau | GB | B |
| 16 | Auswahl Rechteckwerkzeug und Klicken auf Fläche, nachdem anderes Rechteck gezeichnet wurde. | Es entsteht keine Vorschau | GB | B |
| 17 | Auswahl Ellipsenwerkzeug und Klicken auf Fläche, nachdem andere Ellipse gezeichnet wurde. | Es entsteht keine Vorschau | GB | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht. Die Vorschau hat sich beim Klicken (ohne zu ziehen) unerwartet verhalten. Deswegen sind die Testfälle ab Testfall 15 dazugekommen.

Test 8: Klasse: PaintPanel Methode: brush()

Voraussetzungen: gestartete Anwendung, Pinselwerkzeug ausgewählt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bem.** | **Ergebnis** |
| 1 | Klicken und gerade Ziehen mit der Maus auf dem Panel | Es entsteht ein Strich entlang des Mauspfades | GB | B |
| 2 | Klicken und kurven Ziehen mit der Maus auf dem Panel | Es entsteht ein Strich entlang des Mauspfades | GB | B |
| 3 | Klicken und Ziehen mit der Maus außerhalb des Panels und wieder zurück ins Panel | Das Zeichnen bricht nicht ab und wird automatisch fortgesetzt, wenn die Maus wieder im Panel ist. | GB | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 9: Klasse: PaintPanel Methode: line()

Voraussetzungen: gestartete Anwendung, Linienwerkzeug ausgewählt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bem.** | **Ergebnis** |
| 1 | Klicken und Ziehen nach rechts unten mit der Maus auf dem Panel und Loslassen der Maustaste | Es entsteht eine Linie beim Loslassen der Maustaste vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 2 | Klicken und Ziehen nach rechts unten mit der Maus auf dem Panel und Loslassen der Maustaste | Es entsteht eine Linie beim Loslassen der Maustaste vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 3 | Klicken und Ziehen nach rechts unten mit der Maus auf dem Panel und Loslassen der Maustaste | Es entsteht eine Linie beim Loslassen der Maustaste vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 4 | Klicken und Ziehen nach rechts unten mit der Maus auf dem Panel und Loslassen der Maustaste | Es entsteht eine Linie beim Loslassen der Maustaste vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 5 | Klicken und an der gleichen Stelle wieder loslassen | Es entsteht keine Linie. | GB | B |
| 6 | Klicken und dann ziehen außerhalb des Panels und loslassen. | Der Teil der Linie auf dem Panel wird gezeichnet. | GB | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 10: Klasse: PaintPanel Methode: rectangle()

Voraussetzungen: gestartete Anwendung, Rechteckwerkzeug ausgewählt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bem.** | **Ergebnis** |
| 1 | Klicken und Ziehen nach rechts unten mit der Maus auf dem Panel und Loslassen der Maustaste | Es entsteht ein Rechteck beim Loslassen der Maustaste vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 2 | Klicken und Ziehen nach rechts unten mit der Maus auf dem Panel und Loslassen der Maustaste | Es entsteht ein Rechteck beim Loslassen der Maustaste vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 3 | Klicken und Ziehen nach rechts unten mit der Maus auf dem Panel und Loslassen der Maustaste | Es entsteht ein Rechteck beim Loslassen der Maustaste vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 4 | Klicken und Ziehen nach rechts unten mit der Maus auf dem Panel und Loslassen der Maustaste | Es entsteht ein Rechteck beim Loslassen der Maustaste vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 5 | Klicken und an der gleichen Stelle wieder loslassen | Es entsteht kein Rechteck. | GB | B |
| 6 | Klicken und dann ziehen außerhalb des Panels und loslassen. | Der Teil des Rechtecks auf dem Panel wird gezeichnet. | GB | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 11: Klasse: PaintPanel Methode: ellipse()

Voraussetzungen: gestartete Anwendung, Ellipsenwerkzeug ausgewählt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bem.** | **Ergebnis** |
| 1 | Klicken und Ziehen nach rechts unten mit der Maus auf dem Panel und Loslassen der Maustaste | Es entsteht eine Ellipse beim Loslassen der Maustaste vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 2 | Klicken und Ziehen nach rechts unten mit der Maus auf dem Panel und Loslassen der Maustaste | Es entsteht eine Ellipse beim Loslassen der Maustaste vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 3 | Klicken und Ziehen nach rechts unten mit der Maus auf dem Panel und Loslassen der Maustaste | Es entsteht eine Ellipse beim Loslassen der Maustaste vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 4 | Klicken und Ziehen nach rechts unten mit der Maus auf dem Panel und Loslassen der Maustaste | Es entsteht eine Ellipse beim Loslassen der Maustaste vom Startpunkt zur aktuellen Mausposition. | GB | B |
| 5 | Klicken und an der gleichen Stelle wieder loslassen | Es entsteht keine Ellipse. | GB | B |
| 6 | Klicken und dann ziehen außerhalb des Panels und loslassen. | Der Teil der Ellipse auf dem Panel wird gezeichnet. | GB | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 12: Klasse: PaintPanel Methode: erase()

Voraussetzungen: gestartete Anwendung, Radierwerkzeug ausgewählt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bem.** | **Ergebnis** |
| 1 | Klicken und gerade Ziehen mit der Maus auf dem Panel | Es entsteht ein weißer Strich entlang des Mauspfades | GB | B |
| 2 | Klicken und kurven Ziehen mit der Maus auf dem Panel | Es entsteht ein weißer Strich entlang des Mauspfades | GB | B |
| 3 | Klicken und Ziehen mit der Maus außerhalb des Panels und wieder zurück ins Panel | Das Zeichnen bricht nicht ab und wird automatisch fortgesetzt, wenn die Maus wieder im Panel ist. | GB | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 13: Klasse: PaintPanel Methode: save()

Voraussetzungen: gestartete Anwendung, Zeichnung eines Testbildes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bem.** | **Ergebnis** |
| 1 | Klicken auf die Speichern-Schaltfläche in der Menüleiste, Eingeben eines Dateinamens im Dialog, Klicken auf Speichern im Dialog | Es wird ein Bild mit dem Bildnamen im Standardverzeichnis savedPictures gespeichert. | GB | B |
| 2 | Klicken auf die „Speichern unter“-Schaltfläche in der Menüleiste, leerlassen des Dateinamen, Klicken auf speichern im Dialog | Es wird kein Bild gespeichert, der Dialog bleibt sichtbar. | GB | B |
| 3 | Erstellen eines neuen schreibgeschützten Ordners, Klicken auf die „Speichern unter“-Schaltfläche in der Menüleiste, Eingeben eines Dateinamens im Dialog, Klicken auf speichern im Dialog | Es erscheint eine Pop-up-Fehlermeldung: „Fehler beim Speichern des Bildes!“ | GB | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 14: Klasse: PaintPanel Methode: newPanel()

Voraussetzungen: gestartete Anwendung, Zeichnung eines Testbildes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bem.** | **Ergebnis** |
| 1 | Klicken auf die Neu-Schaltfläche in der Menüleiste, Bestätigen der Sicherheitsabfrage, Klicken und Ziehen im Bild | Das Panel wird zunächst komplett weiß, beim Klicken und Ziehen entsteht ein schwarzer Strich (Dicke: 5) entlang des Mauspfades | GB | B |

Bemerkung: Da es hier überlagerte Methoden gibt, wird hier die Methode ohne Parameter getestet. Test 15 testet die Methode mit Parametern. Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 15: Klasse: PaintPanel Methode: newPanel(int width, int height)

Voraussetzungen: gestartete Anwendung, Zeichnung eines Testbildes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bem.** | **Ergebnis** |
| 1 | Klicken auf die „Neu (Blattgröße anpassen)“-Schaltfläche in der Menüleiste, Bestätigen der Sicherheitsabfrage, Eingeben der Breite: 500, Eingeben der Höhe: 500, Klicken und Ziehen im Bild | Das Panel wird zunächst komplett weiß mit einer Größe von 500x500, beim Klicken und Ziehen entsteht ein schwarzer Strich (Dicke: 5) entlang des Mauspfades | GB | B |
| 2 | Wie Nr.1, neue Eingabe: Breite: 500, Höhe: -500 | Es wird kein neues Blatt erstellt und eine Fehlermeldung Ausgegeben, dass keine gültigen Werte eingegeben wurden | GB | B |
| 3 | Wie Nr.1, neue Eingabe: Breite: -500, Breite: 500 | Es wird kein neues Blatt erstellt und eine Fehlermeldung Ausgegeben, dass keine gültigen Werte eingegeben wurden | GB | B |
| 4 | Wie Nr.1, neue Eingabe: Breite: 500, Höhe: 0 | Es wird kein neues Blatt erstellt und eine Fehlermeldung Ausgegeben, dass keine gültigen Werte eingegeben wurden | GB | B |
| 5 | Wie Nr.1, neue Eingabe: Breite: 0, Breite: 500 | Es wird kein neues Blatt erstellt und eine Fehlermeldung Ausgegeben, dass keine gültigen Werte eingegeben wurden | GB | B |
| 6 | Wie Nr.1, neue Eingabe: Breite: 500, Höhe: \*leer\* | Es wird kein neues Blatt erstellt und eine Fehlermeldung Ausgegeben, dass keine gültigen Werte eingegeben wurden | GB | B |
| 7 | Wie Nr.1, neue Eingabe: Breite: \*leer\*, Breite: 500 | Es wird kein neues Blatt erstellt und eine Fehlermeldung Ausgegeben, dass keine gültigen Werte eingegeben wurden | GB | B |
| 8 | Wie Nr.1, neue Eingabe: Breite: 500, Höhe: abc | Es wird kein neues Blatt erstellt und eine Fehlermeldung Ausgegeben, dass keine gültigen Werte eingegeben wurden | GB | B |
| 9 | Wie Nr.1, neue Eingabe: Breite: abc, Breite: 500 | Es wird kein neues Blatt erstellt und eine Fehlermeldung Ausgegeben, dass keine gültigen Werte eingegeben wurden | GB | B |
| 10 | Wie Nr.1, neue Eingabe: Breite: 30000, Höhe 30000 | Das Panel wird zunächst komplett weiß mit einer Größe von 30000x30000, beim Klicken und Ziehen entsteht ein schwarzer Strich (Dicke: 5) entlang des Mauspfades | ZB | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 16: Klasse: Frame Methode: Frame() (Konstruktor)

Voraussetzungen: -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bem.** | **Ergebnis** |
| 1 | Starten der Anwendung, auf den Button „Verkleinern“ in der Titelleiste drücken, das Fenster kleiner als die weiße Zeichenfläche ziehen, auf den Button „Beenden“ in der Titelleiste drücken | Es öffnet sich ein maximiertes Fenster mit einer weißen Zeichenfläche. Beim Verkleinern des Fensters wird das Fenster halb so groß wie der Bildschirm. Beim verkleinern des Fensters (kleiner als Zeichenfläche) entstehen Scrollbars vertikal und horizontal. Beim Klick auf das Kreuz wird das Programm beendet | GB | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 17: Klasse: Frame Methode: createMenuBar()

Voraussetzungen: -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bem.** | **Ergebnis** |
| 1 | Starten der Anwendung | Es erscheint am oberen Rand des Fensters eine Menüleiste mit den Reitern „Datei“ und „Werkzeuge“.  Menü „Datei“ enthält: „Neu“, „Neu (Blattgröße anpassen)“, „Laden“, „Speichern“, „Speichern unter“ und „Beenden“. Menü „Werkzeuge“ enthält: „Pinsel“, „Linie“, „Viereck“, „Ellipse“ und „Radierer“. | GB | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 18: Klasse: Frame Methode: createSymbolBar()

Voraussetzungen: -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bem.** | **Ergebnis** |
| 1 | Starten der Anwendung | Es erscheint am oberen Rand unter der Menüleiste eine Symbolleiste. Von links nach rechts auf der Symbolleiste: Pinsel, Linie, Viereck, Ellipse, Radierer, Strichstärke (5,0), Farbauswahl. | GB | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 19: Klasse: Frame Methode: createButton()

Voraussetzungen: gestartete Anwendung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bem.** | **Ergebnis** |
| 1 | Starten der Anwendung | Button „Pinsel und „Schwarz“ sind sichtbar ausgewählt | GB | B |
| 2 | Kurzes Schweben mit der Maus über dem Linien-Button, dann anklicken. | Es erscheint der Tooltip „Linie“, danach ist dieser Button sichtbar ausgewählt. | GB | B |
| 3 | Kurzes Schweben mit der Maus über dem Viereck-Button, dann anklicken. | Es erscheint der Tooltip „Viereck“, danach ist dieser Button sichtbar ausgewählt. | GB | B |
| 4 | Kurzes Schweben mit der Maus über dem Ellipse-Button, dann anklicken. | Es erscheint der Tooltip „Ellipse“, danach ist dieser Button sichtbar ausgewählt. | GB | B |
| 5 | Kurzes Schweben mit der Maus über dem Radierer-Button, dann anklicken. | Es erscheint der Tooltip „Radierer“, danach ist dieser Button sichtbar ausgewählt. | GB | B |
| 6 | Anklicken des Linien Buttons, kurzes Schweben mit der Maus über dem Pinsel-Button, dann anklicken. | Der Button „Linie“ ist sichtbar ausgewählt. Es erscheint der Tooltip „Pinsel“, danach ist dieser Button sichtbar ausgewählt. | GB | B |
| 7 | Kurzes Schweben mit der Maus über dem Rot-Button, dann anklicken. | Es erscheint der Tooltip „rot“, danach ist dieser Button sichtbar ausgewählt. | GB | B |
| 8 | Kurzes Schweben mit der Maus über dem Blau-Button, dann anklicken. | Es erscheint der Tooltip „blau“, danach ist dieser Button sichtbar ausgewählt. | GB | B |
| 9 | Kurzes Schweben mit der Maus über dem Gelb-Button, dann anklicken. | Es erscheint der Tooltip „gelb“, danach ist dieser Button sichtbar ausgewählt. | GB | B |
| 10 | Kurzes Schweben mit der Maus über dem Weiß-Button, dann anklicken. | Es erscheint der Tooltip „weiß“, danach ist dieser Button sichtbar ausgewählt. | GB | B |
| 11 | Kurzes Schweben mit der Maus über dem Cyan-Button, dann anklicken. | Es erscheint der Tooltip „cyan“, danach ist dieser Button sichtbar ausgewählt. | GB | B |
| 12 | Kurzes Schweben mit der Maus über dem Grün-Button, dann anklicken. | Es erscheint der Tooltip „grün“, danach ist dieser Button sichtbar ausgewählt. | GB | B |
| 13 | Kurzes Schweben mit der Maus über dem Magenta-Button, dann anklicken. | Es erscheint der Tooltip „magenta“, danach ist dieser Button sichtbar ausgewählt. | GB | B |
| 14 | Kurzes Schweben mit der Maus über dem Orange-Button, dann anklicken. | Es erscheint der Tooltip „orange“, danach ist dieser Button sichtbar ausgewählt. | GB | B |
| 15 | Kurzes Schweben mit der Maus über dem Pink-Button, dann anklicken. | Es erscheint der Tooltip „pink“, danach ist dieser Button sichtbar ausgewählt. | GB | B |
| 16 | Kurzes Schweben mit der Maus über dem Hellgrau-Button, dann anklicken. | Es erscheint der Tooltip „hellgrau“, danach ist dieser Button sichtbar ausgewählt. | GB | B |
| 17 | Kurzes Schweben mit der Maus über dem Grau-Button, dann anklicken. | Es erscheint der Tooltip „grau“, danach ist dieser Button sichtbar ausgewählt. | GB | B |
| 18 | Kurzes Schweben mit der Maus über dem Dunkelgrau-Button, dann anklicken. | Es erscheint der Tooltip „dunkelgrau“, danach ist dieser Button sichtbar ausgewählt. | GB | B |
| 19 | Anklicken des Rot-Buttons, kurzes Schweben mit der Maus über dem Schwarz-Button, dann anklicken. | Der Button „Rot“ ist sichtbar ausgewählt. Es erscheint der Tooltip „schwarz“, danach ist dieser Button sichtbar ausgewählt. | GB | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 20: Klasse: Frame.ButtonAndTextfieldListener Methode: actionPerformed()

Voraussetzungen: gestartete Anwendung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bem.** | **Ergeb-nis** |
| 1 | Zeichnen eines Pinselstrichs, Auswahl Radierer, Radieren, Wechsel auf den Pinsel, zeichnen | Schwarzer Strich wird gezeichnet, Radierer ist ausgewählt, es wird radiert, Pinsel ist ausgewählt, man zeichnet schwarz | GB | B |
| 2 | Auswahl rot und Linie, Zeichnen einer Linie, Auswahl Radierer, Radieren, Wechsel auf die Linie, zeichnen | Rote Linie wird gezeichnet, Radierer ist ausgewählt, es wird radiert, Linie ist ausgewählt, man zeichnet rote Linien | GB | B |
| 3 | Auswahl blau und Viereck, Zeichnen eines Vierecks, Auswahl Radierer, Radieren, Wechsel auf das Viereck, zeichnen | Blaues Viereck wird gezeichnet, Radierer ist ausgewählt, es wird radiert, Viereck ist ausgewählt, man zeichnet blaue Vierecke | ZB | B |
| 4 | Auswahl gelb und Ellipse, Zeichnen einer Ellipse, Auswahl Radierer, Radieren, Wechsel auf die Ellipse, zeichnen | Gelbe Ellipse wird gezeichnet, Radierer ist ausgewählt, es wird radiert, Ellipse ist ausgewählt, man zeichnet gelbe Ellipsen | ZB | B |
| 5 | Auswahl cyan und Viereck, Zeichnen eines Vierecks, Auswahl Radierer, Radieren, erneutes Klicken auf den Button Radierer, Wechsel auf Ellipse, zeichnen | Cyan Viereck wird gezeichnet, Radierer ist ausgewählt, es wird radiert, Ellipse ist ausgewählt, man zeichnet cyan Ellipsen | C2, zum Testen, ob die Farbe des Vierecks auch bei mehrmaligem drücken des Radier-Buttons funktioniert | B |
| 6 | Auswahl Pinsel, zeichnen, Eingabe Strichstärke 10, bestätigen mit Enter, zeichnen | Pinselstrich mit Stärke 5, Textfeld zeigt Strichstärke 10.0 und verliert Fokus, Pinselstrich mit Stärke 10 | GB | B |
| 7 | Auswahl Pinsel, zeichnen, Eingabe Strichstärke 0, bestätigen mit Enter, zeichnen | Pinselstrich mit Stärke 5, Textfeld zeigt Strichstärke 5.0 und verliert Fokus, Pinselstrich mit Stärke 5 | GB | B |
| 8 | Auswahl Pinsel, zeichnen, Eingabe Strichstärke -1, bestätigen mit Enter, zeichnen | Pinselstrich mit Stärke 5, Textfeld zeigt Strichstärke 5.0 und verliert Fokus, Pinselstrich mit Stärke 5 | GB | B |
| 9 | Auswahl Pinsel, zeichnen, Eingabe Strichstärke „abc“ , bestätigen mit Enter, zeichnen | Pinselstrich mit Stärke 5, Textfeld zeigt Strichstärke 5.0 und verliert Fokus, Pinselstrich mit Stärke 5 | GB | B |
| 10 | Auswahl Pinsel, zeichnen, Eingabe Strichstärke 0,1, bestätigen mit Button „OK“, zeichnen | Pinselstrich mit Stärke 0,1, Textfeld zeigt Strichstärke 0,1 und verliert Fokus, Pinselstrich mit Stärke 0,1 | GB | B |
| 11 | Auswahl Pinsel, zeichnen, Auswahl Radierer, radieren, Auswahl blau, radieren, Auswahl Pinsel, zeichnen | Schwarze Linie zeichnen, Linie radieren, nach Farbauswahl nochmal radieren, mit Pinsel schwarze Linie zeichnen | GB, zum Testen, ob beim Radieren die Farbauswahl nicht funktioniert | B |
| 12 | Auswahl Pinsel, Auswahl <Farbe>, zeichnen, Auswahl Radierer, radieren, Auswahl Pinsel, zeichnen (Durchführen mit jeder Farbe) | Linie in <Farbe> wird gezeichnet, dann radiert (der Farbbutton ist nicht mehr angewählt) und anschließend in der gleichen Farbe wieder gezeichnet (der Farbbutton ist wieder angewählt) | C2, aus Gründen der Übersichtlichkeit wurden alle Testfälle (Farben) in einem Punkt zusammengefasst | B |
| 13 | Zeichnen eines Testbildes, klicken auf die Schaltfläche „Neu“ im Reiter „Datei“, im Dialog klicken auf nein | Die Sicherheitsabfrage erscheint, das Testbild bleibt erhalten | GB | B |
| 14 | Zeichnen eines Testbildes, klicken auf die Schaltfläche „Neu“ im Reiter „Datei“, im Dialog klicken auf ja | Die Sicherheitsabfrage erscheint, es erscheint ein gleichgroßes leeres (weißes) Blatt | GB | B |
| 15 | Zeichnen eines Testbildes, klicken auf die Schaltfläche „Neu (Blattgröße anpassen)“ im Reiter „Datei“, im Dialog klicken auf nein | Die Sicherheitsabfrage erscheint, das Testbild bleibt erhalten | GB | B |
| 16 | Zeichnen eines Testbildes, klicken auf die Schaltfläche „Neu (Blattgröße anpassen)“ im Reiter „Datei“, im Dialog klicken auf ja, Eingeben der Breite: 500, Eingeben der Höhe: 500 | Die Sicherheitsabfrage erscheint, es erscheint ein leeres (weißes) Blatt in der Größe 500x500 | GB, andere Eingaben wurden bereits in Test 15 getestet | B |
| 17 | klicken auf die Schaltfläche „Beenden“ im Reiter „Datei“, im Dialog klicken auf nein | Die Sicherheitsabfrage erscheint, es bleibt bei der Situation davor | GB | B |
| 18 | klicken auf die Schaltfläche „Beenden“ im Reiter „Datei“, im Dialog klicken auf ja | Die Sicherheitsabfrage erscheint, das Programm beendet sich | GB | B |
| 19 | Löschen des Ordners „savedPictures“, klicken auf die Schaltfläche „speichern“ im Reiter „Datei“, klicken auf „abbrechen“ | Der Speichern-Dialog erscheint, es wird nicht gespeichert und kein Ordner erstellt | GB | B |
| 20 | Löschen des Ordners „savedPictures“, klicken auf die Schaltfläche „speichern“ im Reiter „Datei“, klicken auf ja, Eingeben des Dateinamens „Test1“ | Der Speichern-Dialog erscheint, der Ordner „savedPictures“ wird erstellt, nach dem Eingeben des Dateinamens wird die Datei „Test1.jpg“ im erstellten Ordner gespeichert | GB | B |
| 21 | direkt im Anschluss an Testfall 20: zeichnen, klicken auf die Schaltfläche „speichern“ im Reiter „Datei“ | Der Speichern-Dialog erscheint, die Datei „Test1.jpg“ wird im erstellten Ordner überschrieben | GB | B |
| 22 | Neues Blatt erstellen, klicken auf die Schaltfläche „speichern“ im Reiter „Datei“, Eingeben des Dateinamens „Test2.jpg“ | Der Speichern-Dialog erscheint, die Datei „Test2.jpg“ wird im erstellten Ordner gespeichert | GB | B |
| 23 | Löschen des Ordners „savedPictures“, klicken auf die Schaltfläche „speichern unter“ im Reiter „Datei“, klicken auf „abbrechen“ | Der Speichern-Dialog erscheint, es wird nicht gespeichert | GB | B |
| 24 | Löschen des Ordners „savedPictures“, klicken auf die Schaltfläche „speichern unter“ im Reiter „Datei“, Eingeben des Dateinamens „Test3“ | Der Speichern-Dialog erscheint, der Ordner „savedPictures“ wird erstellt, nach dem Eingeben des Dateinamens wird die Datei „Test3.jpg“ im erstellten Ordner gespeichert | GB | B |
| 25 | direkt im Anschluss an Testfall 24: zeichnen, klicken auf die Schaltfläche „speichern unter“ im Reiter „Datei“, Eingeben des Dateinamens „Test4“ | Der Speichern-Dialog erscheint, nach dem Eingeben des Dateinamens wird die Datei „Test4.jpg“ im erstellten Ordner gespeichert | GB | B |
| 26 | Klicken auf die Schaltfläche „speichern unter“ im Reiter „Datei“, Eingeben des Dateinamens „Test5.jpg“ | Der Speichern-Dialog erscheint, die Datei „Test5.jpg“ wird im erstellten Ordner gespeichert | GB | B |
| 27 | Neues Blatt erstellen, Klicken auf die Schaltfläche „laden“ im Reiter „Datei“, klicken auf nein | Die Sicherheitsabfrage erscheint, es wird nichts geladen | GB | B |
| 28 | Neues Blatt erstellen, Klicken auf die Schaltfläche „laden“ im Reiter „Datei“, klicken auf ja, im Dialog wird die Datei „Test4.jpg“ ausgewählt, zeichnen | Es erscheint ein leeres Blatt, die Sicherheitsabfrage erscheint, das gespeicherte Bild erscheint im Panel und kann bearbeitet werden | GB | B |
| 29 | Direkt im Anschluss an Testfall 28: Klicken auf die Schaltfläche „speichern“ im Reiter „Datei“ | Die Sicherheitsabfrage erscheint, die Datei „Test4.jpg“ wird (ohne Speicherdialog) überschrieben | GB, zum Prüfen, ob der Speicherdialog ausbleibt | B |
| 30 | Drücken <Taste> (Hotkey), testen der zugehörigen Funktion | <Funktion> wird ausgewählt und kann genutzt werden | ZB, aus Gründen der Übersichtlichkeit wurden alle Testfälle (Hotkeys) in einem Punkt zusammengefasst, eine Liste aller Hotkeys ist im Anschluss an die Tests aufgelistet | B |
| 31 | Drücken des Buttons für <Werkzeug> in der Menüleiste im Reiter „Werkzeuge“, testen des zugehörigen Werkzeugs | <Werkzeug> ist ausgewählt und kann genutzt werden | ZB, aus Gründen der Übersichtlichkeit wurden alle Testfälle (Werkzeuge) in einem Punkt zusammengefasst | B |
| 32 | Drücken der Taste F10, navigieren mit den Pfeiltasten durch die Menüleiste | Menüleiste ist mit Tastatur navigierbar | ZB | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist nicht komplett erreicht, da nicht getestet werden konnte, was passiert, wenn das Standardverzeichnis für eigene Bilder nicht erfolgreich erstellt werden konnte.

Test 21: Klasse: Frame.MouseListener Methode: mousePressed()

Voraussetzungen: gestartete Anwendung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bem.** | **Ergebnis** |
| 1 | Auswahl Pinsel, drücken der linken Maustaste im Panel (ohne zu ziehen oder loszulassen) | Es entsteht ein schwarzer Punkt an der Stelle, an der die linke Maustaste gedrückt wurde | GB | B |
| 2 | Auswahl Pinsel, zeichnen, Auswahl Radierer drücken der linken Maustaste im Panel an einer nicht weißen Stelle (ohne zu ziehen oder loszulassen) | Es entsteht ein weißer Punkt an der Stelle, an der die linke Maustaste gedrückt wurde | GB | B |
| 3 | Auswahl Pinsel, drücken der rechten Maustaste im Panel (ohne zu ziehen oder loszulassen) | Es ändert sich nichts | GB | B |
| 4 | Auswahl Pinsel, drücken der mittleren Maustaste im Panel (ohne zu ziehen oder loszulassen) | Es ändert sich nichts | ZB | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 22: Klasse: Frame.MouseListener Methoden: mouseReleased()

mouseDragged()

Voraussetzungen: gestartete Anwendung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bem.** | **Ergebnis** |
| 1 | Auswahl Linie, drücken der rechten Maustaste im Panel, ziehen und wieder loslassen | Es ändert sich nichts | GB | B |
| 2 | Auswahl Viereck, drücken der rechten Maustaste im Panel, ziehen und wieder loslassen | Es ändert sich nichts | GB | B |
| 3 | Auswahl Ellipse, drücken der rechten Maustaste im Panel, ziehen und wieder loslassen | Es ändert sich nichts | GB | B |

Bemerkung: Das Zeichnen der Figuren, mit Pinsel und Radierer wurden bereits in anderen Tests getestet und wird hier nicht wiederholt. Aus funktionsgründen ist es nur möglich beide Funktionen zusammen zu testen. Vollständige Abdeckung nach C2-Verfahren ist erreicht.

Test 23: Klasse: PaintProgram Methode: main()

Voraussetzungen: -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nr.** | **Testfall** | **Erwartetes Ergebnis** | **Bem.** | **Ergebnis** |
| 1 | Starten der Anwendung | Das Zeichenprogramm startet | GB | B |

Bemerkung: Vollständige Abdeckung nach C2-Verfahren ist erreicht.

PITest:

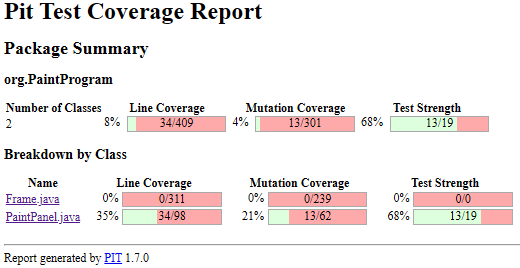


Abbildung Ergebnis PITest

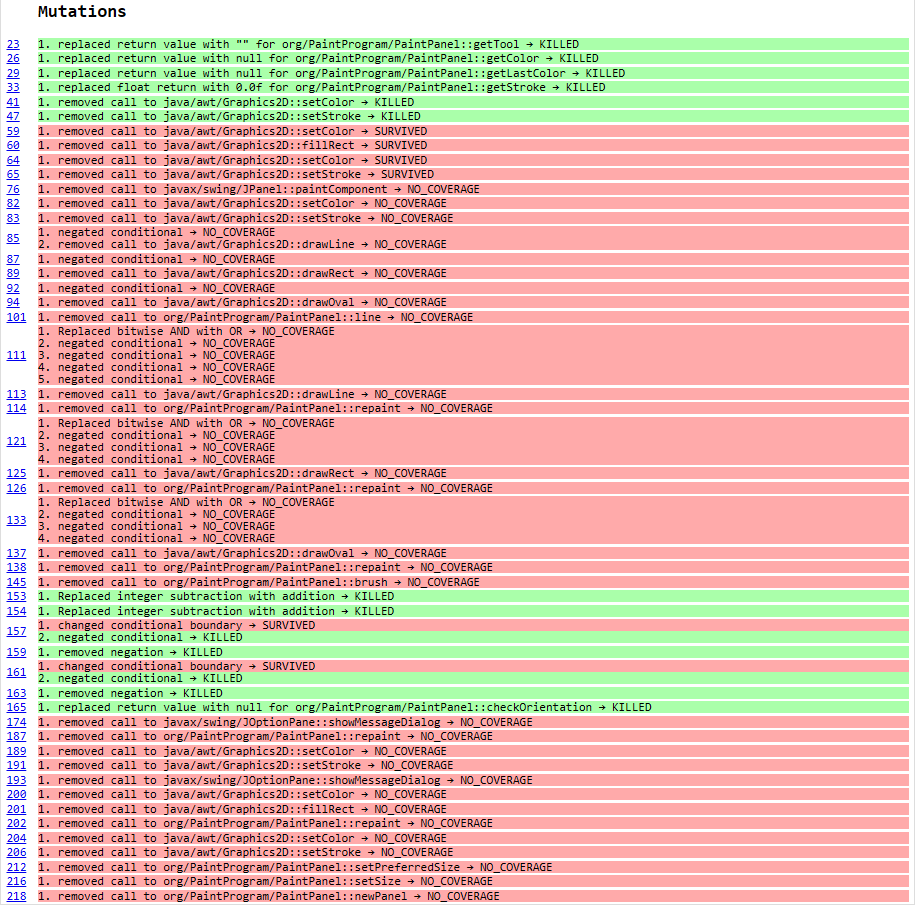


Abbildung Ergebnisse des PITest in PaintPanel

Die Klasse Frame hat keine automatisierten Tests und hat damit eine Rate von 0%. In der Klasse PaintPanel sind alle roten Zeilen ebenfalls durch die automatisierten Tests nicht abgedeckt oder enthalten grafische Elemente, die mit dem PITest nicht sinnvoll getestet werden können. Alle anderen Fälle finden sich in den grünen Zeilen und haben bestanden.

**4. Anhang**

Hotkeys:

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Funktion** | **Hotkey** |
| 1 | Steuerung Menüleiste mit Tastatur (Navigation mit Pfeiltasten, Bestätigen mit Enter, Beenden mit ESC) | F10 |
| 2 | Pinsel | Alt + P |
| 3 | Linie | Alt + L |
| 4 | Viereck | Alt + V |
| 5 | Ellipse | Alt + E |
| 6 | Radierer | Alt + X |
| 7 | Strichstärke: Textfeld mit Tastatur anwählen | Alt + D |
| 8 | Neu | Strg + N |
| 9 | Neu (Blattgröße anpassen) | Strg + Umschalt + N |
| 10 | Laden | Strg + L |
| 11 | Speichern | Strg + S |
| 12 | Speichern unter | Strg + Umschalt + S |
| 13 | Beenden (Windows Shortcut und nicht im Programm) | Alt + F4 |
| 14 | Farbe 1 | Alt + 1 |
| 15 | Farbe 2 | Alt + 2 |
| 16 | Farbe 3 | Alt + 3 |
| 17 | Farbe 4 | Alt + 4 |
| 18 | Farbe 5 | Alt + 5 |
| 19 | Farbe 6 | Alt + 6 |
| 20 | Farbe 7 | Alt + 7 |
| 21 | Farbe 8 | Alt + 8 |
| 22 | Farbe 9 | Alt + 9 |
| 23 | Farbe 10 | Alt + 0 |
| 24 | Farbe 11 | Alt + Numpad1 |
| 25 | Farbe 12 | Alt + Numpad2 |
| 26 | Farbe 13 | Alt + Numpad3 |

Abbildungen vom Programm:

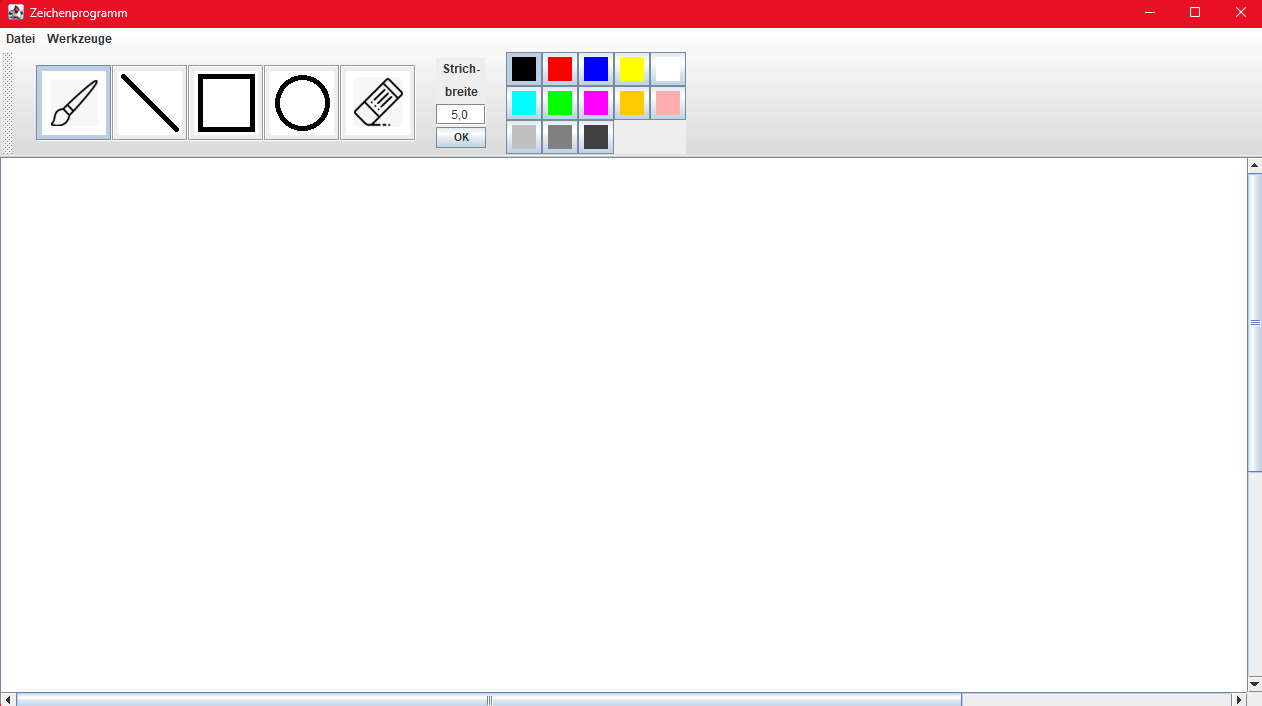


Abbildung Zeichenprogramm nach Programmstart

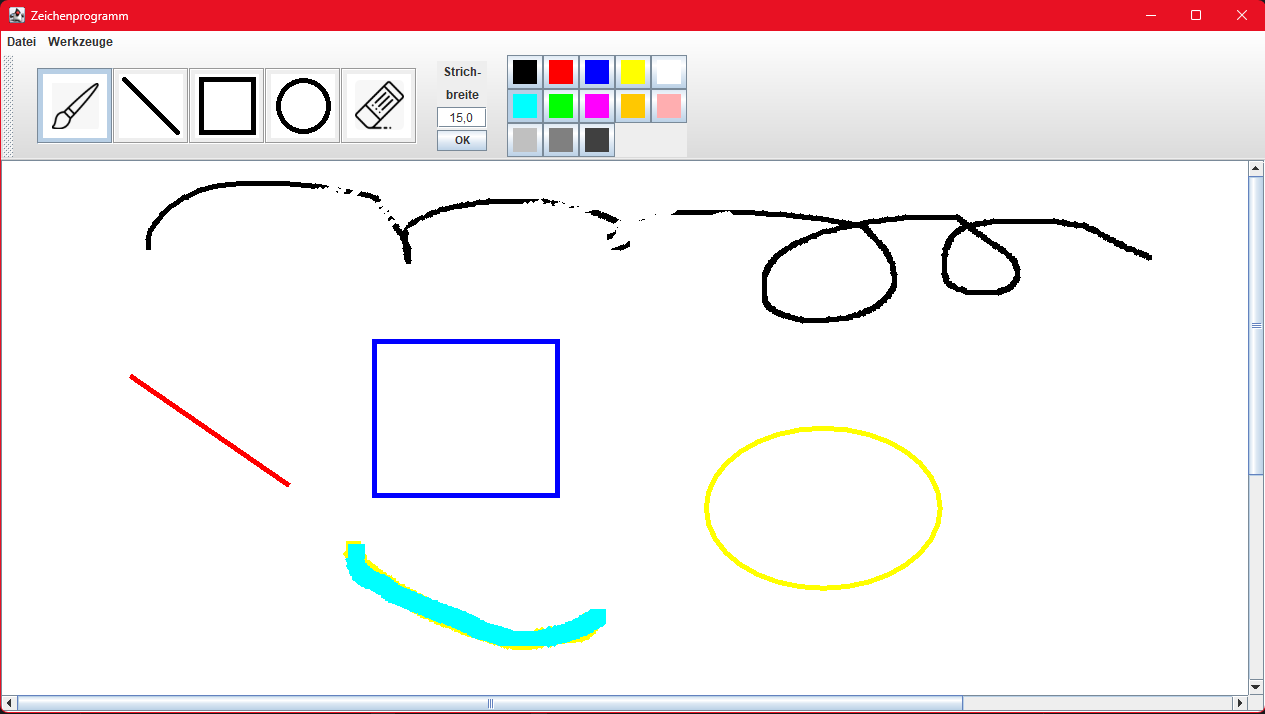


Abbildung Zeichenprogramm mit genutzten Zeichenwerkzeugen