Über die Eignung der Programmiersprache Scratch zur Aneignung von Programmierungskompetenzen

Eine Studie bei Mathematiklehramtsstudierenden

Daniel Walter

Arbeitskreis Mathematikunterricht und Informatik in der GDM

29.09.2013

Inhalt

1. **Scratch: Theoretische Grundlagen**
   - Motivation und Ziele von Scratch
   - Wesen und Funktionsweise
   - Forschungsstand
2. **Empirische Studie**
   - Begründung und Ziel des empirischen Vorhabens
   - Forschungsdesign
   - Methoden der Datenerhebung- und Auswertung
   - Darstellung und Interpretation der Daten
3. **Schlussfolgerungen**
   - Erkenntnisse aus der Untersuchung
   - Umgang mit Scratch

Inhalt

1. **Scratch: Theoretische Grundlagen**
   - Motivation und Ziele von Scratch
   - Wesen und Funktionsweise
   - Forschungsstand
2. **Empirische Studie**
   - Begründung und Ziel des empirischen Vorhabens
   - Forschungsdesign
   - Methoden der Datenerhebung- und Auswertung
   - Darstellung und Interpretation der Daten
3. **Schlussfolgerungen**
   - Erkenntnisse aus der Untersuchung
   - Umgang mit Scratch

Motivation und Ziele

Motivation für die Entwicklung

- Anlehnung an Ideen LOGOs
- Nachteile bisheriger Programmiersprachen beheben
- Idee: „LEGO-Steine“

Motivation und Ziele

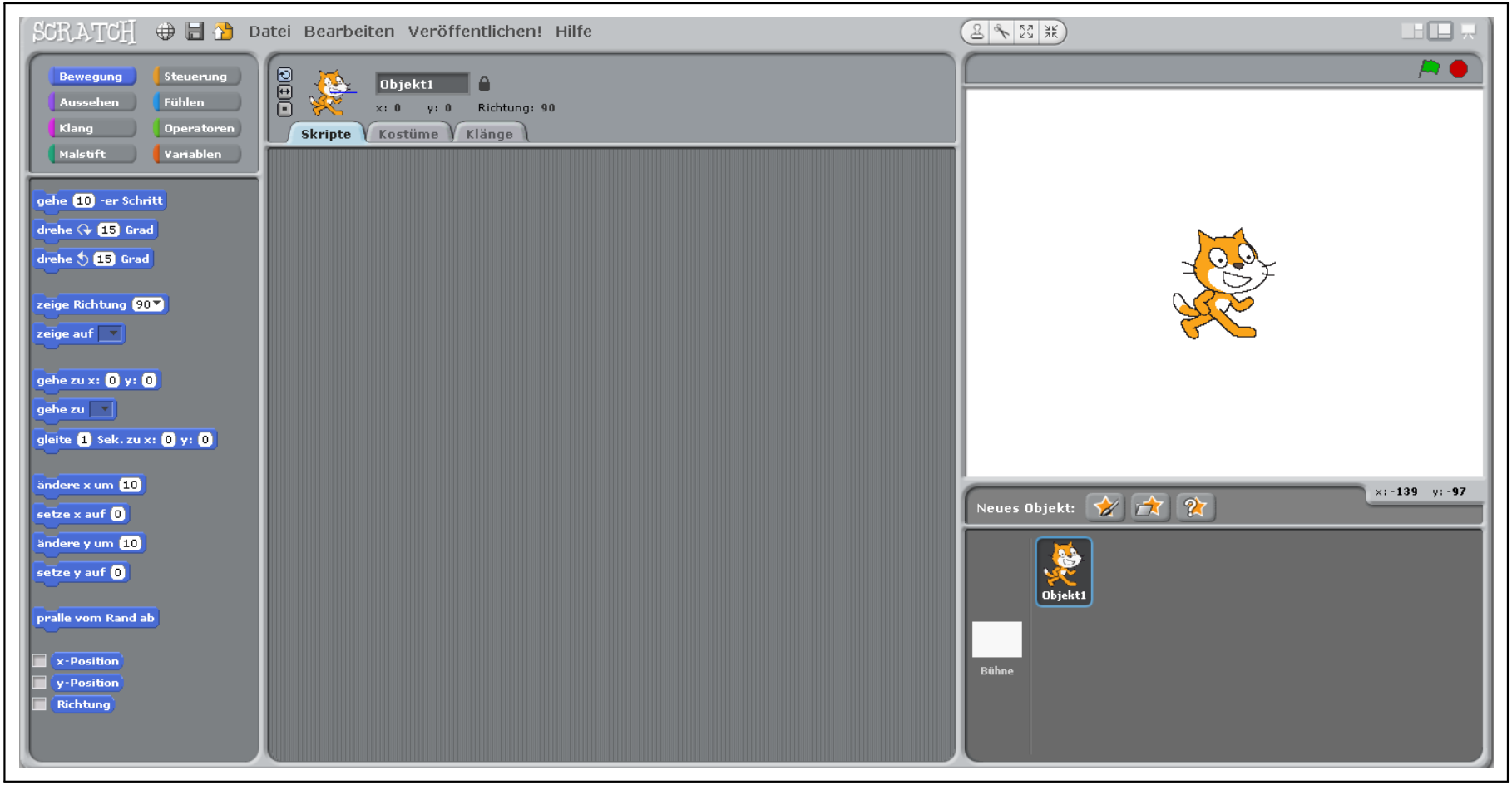
Motivation für die Entwicklung

- Anlehnung an Ideen LOGOs
- Nachteile bisheriger Programmiersprachen beheben
- Idee: „LEGO-Steine“

Zentrale Ziele der Entwickler

- Spielerischer Umgang mit Aspekten der Programmierung
- Eigene Ideen verwirklichen
- Sozialer Aspekt: Ideenaustausch über Scratch-Projekte

Programmieroberfläche



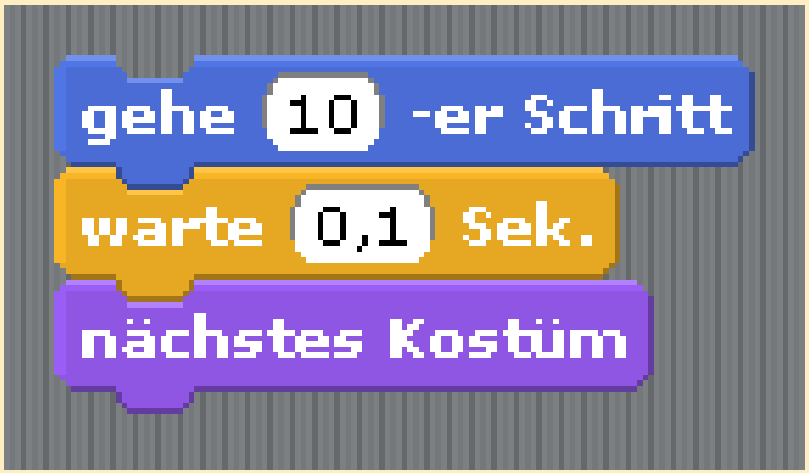
Drag and Drop: Vermeidung von Syntaxfehlern

Beispiel A

Beispiel B

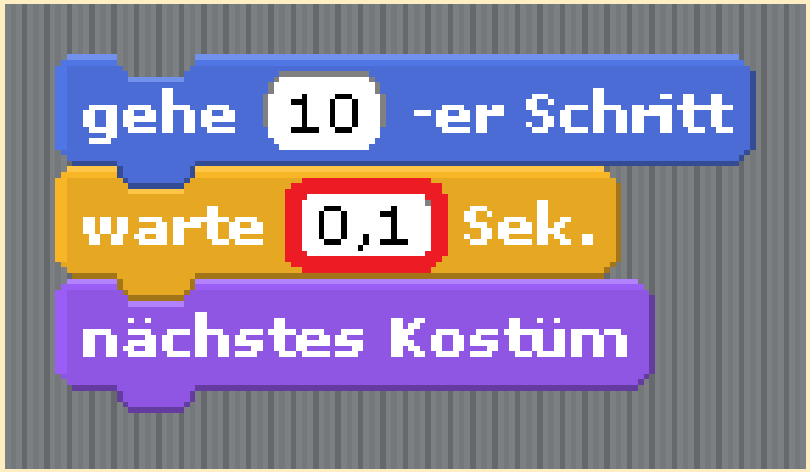
Vermeidung von Semantikfehlern

Eingabe



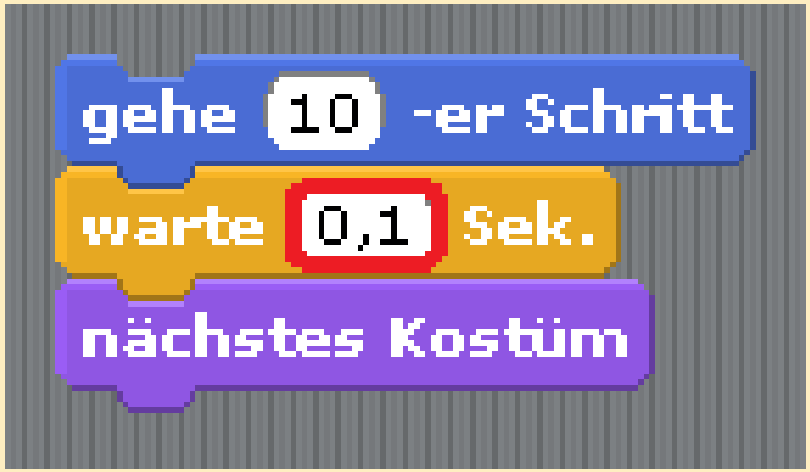
Vermeidung von Semantikfehlern

Eingabe

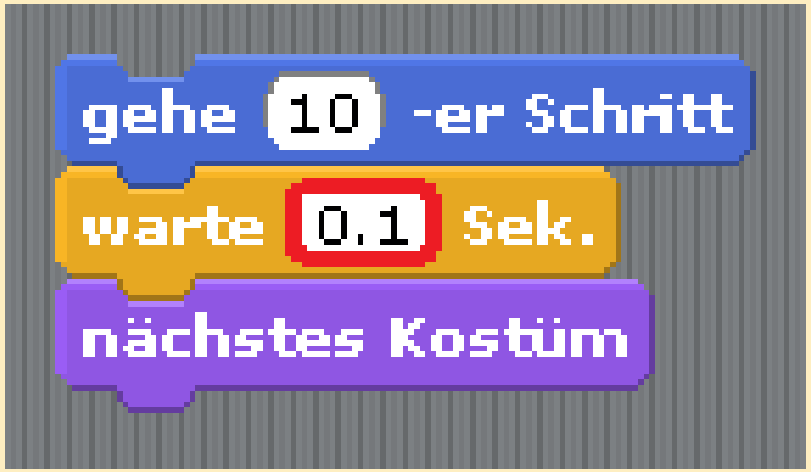


Vermeidung von Semantikfehlern

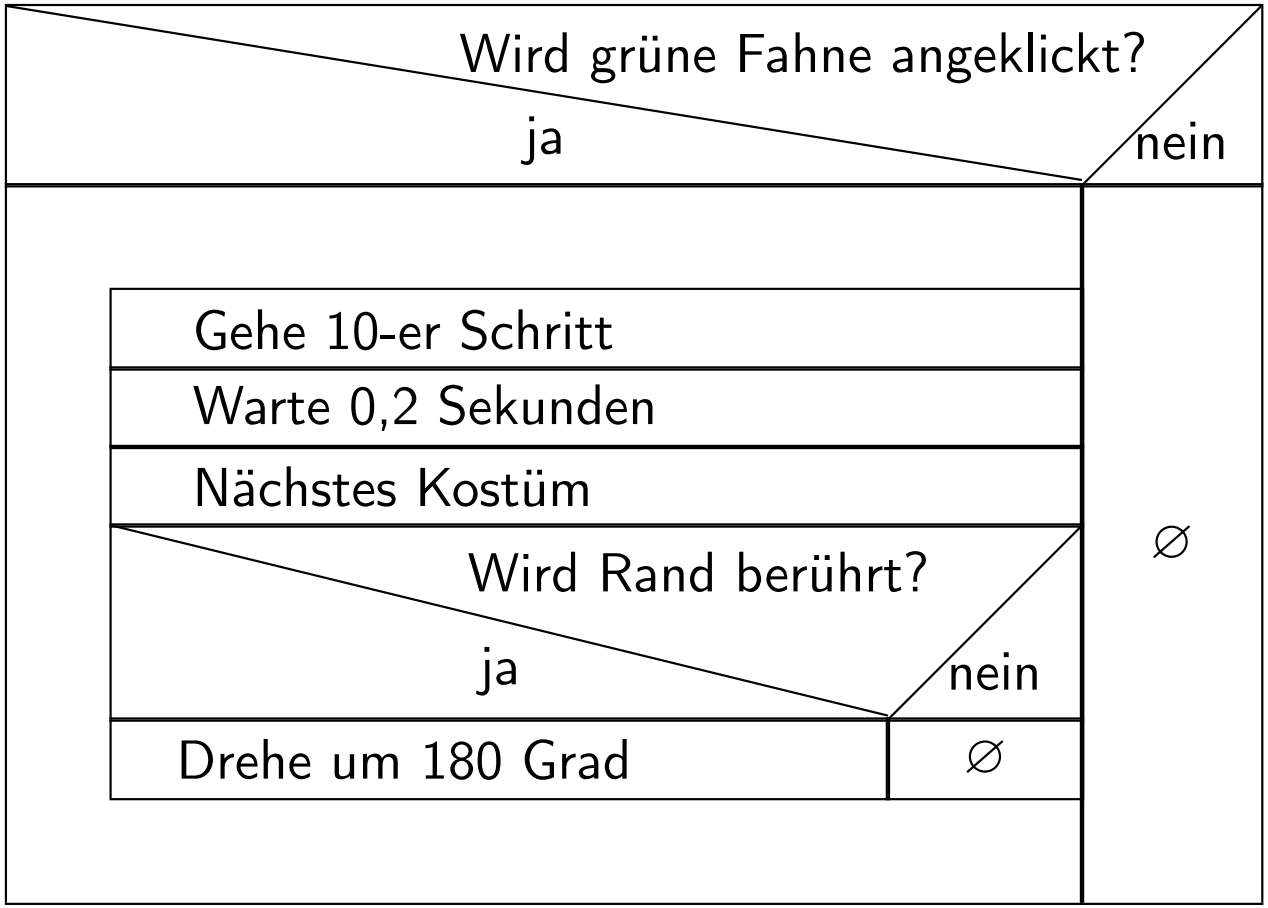
Eingabe



Veränderte Ausgabe



Ähnlichkeit: Scratch-Codes und Struktogramme



Forschungsansätze zu Scratch

Computer Clubhouse der Scratch-Entwickler

Grundlegende Konzepte der Programmierung werden erlernt durch:

- Eigentätige, spielerische Aktivitäten am PC
- Abwesenheit programmierungserfahrener Mentoren

Forschungsansätze zu Scratch

Computer Clubhouse der Scratch-Entwickler

Grundlegende Konzepte der Programmierung werden erlernt durch:

- Eigentätige, spielerische Aktivitäten am PC
- Abwesenheit programmierungserfahrener Mentoren

Scratch in der Schule

- Projekt: InfoSphere - Schülerlabor Informatik (RWTH Aachen)
- Unterrichtsmaterialien: „Scratch-Modul“

Inhalt

1. Scratch: Theoretische Grundlagen
   - Motivation und Ziele von Scratch
   - Wesen und Funktionsweise
   - Forschungsstand
2. **Empirische Studie**
   - Begründung und Ziel des empirischen Vorhabens
   - Forschungsdesign
   - Methoden der Datenerhebung- und Auswertung
   - Darstellung und Interpretation der Daten
3. Schlussfolgerungen
   - Erkenntnisse aus der Untersuchung
   - Umgang mit Scratch

Argumente: Programmierung in der Schule

Lernpotential der Programmierung

- Prozesshaftigkeit des Lernens: Von der Idee zum Algorithmus
- kreatives und logisches Denken
- verschiedene Zugangsweisen zu einem Problem

Weitere Argumente:

- Schnittstellen zur Mathematik
- Propädeutische Funktion

Vernachlässigung der Programmierung

Resultierende Probleme

- Fehlende Vermittlung von Schlüsselkompetenzen
- Abhängigkeit von Bildung bzw. neuen Technologien
- Fachkräftemangel im MINT-Bereich

Ziele des empirischen Vorhabens

Können bzw. wollen angehende Lehrkräfte die Programmierung in der Schule unterrichten?

- Programmierungskompetenzen der Lehrenden
- Einstellung der Lehrenden zur Programmierung und zum PC-Einsatz

Einfluss der Programmiersprache Scratch

Eignet sich Scratch für ...

- die Aneignung von Programmierungskompetenzen?
- die Beeinflussung der Einstellungen angehender Lehrkräfte?

Forschungsdesign

Rahmenbedingungen

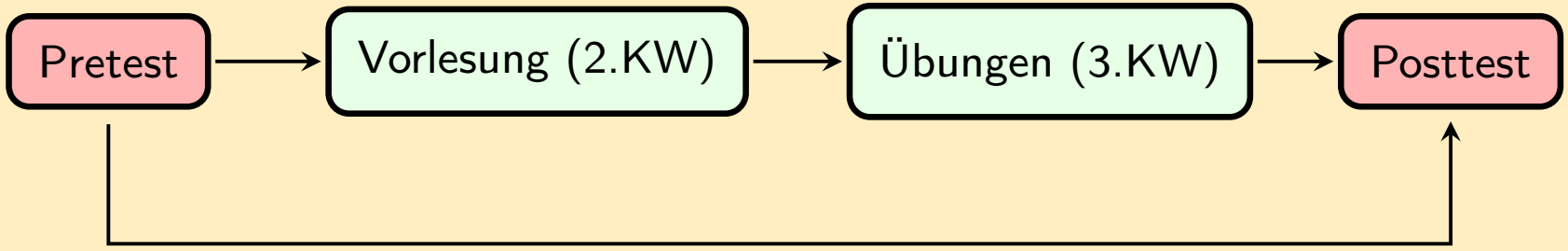
- Universität Vechta
- Lehrveranstaltung im BA „Combined Studies“ - Mathematik
- 88 Probanden
- Panelstudie: (Pre- und Posttest)
- Erhebungszeitraum: 2. und 3. KW 2013

Vorerfahrungen der Probanden

- Grundkenntnisse der Programmierung und PC-Anwendung aus bisherigen Lehrveranstaltungen

Forschungsdesign

Untersuchungszeitraum



Gestaltung der Tests

Kombination:

Item Batterien - Likert Skala

- Einstellung bzgl. der Programmierung
- Einstellung bzgl. der PC-Nutzung in der Schule
- Einstellung bzgl. Scratch (Posttest)

Aufgaben zum Programmierungsverständnis

- Kompaktes, effizientes Programmieren
- Korrektur logischer Fehler in Algorithmen

Aufgabe 1: Kompaktes, effizientes Programmieren

$a \leftarrow 3$
$e \leftarrow a + 2$
$a \leftarrow \frac{e}{2} + 1$
$m \leftarrow 5 \cdot 8$
$x \leftarrow m \cdot 2$
$m \leftarrow m + e$
$e \leftarrow e - 1$
$m \leftarrow m + e$
$e \leftarrow e - 1$
$m \leftarrow m + e$
$e \leftarrow e - 1$
$t \leftarrow x + 3$
Gib aus: m

- Fiktiver Algorithmus ohne Kontext
- Aufgabe: Algorithmus kompakt aufbereiten
- Nutzbarmachung von Strukturelementen

Aufgabe 1: Kompaktes, effizientes Programmieren

$a \leftarrow 3$
$e \leftarrow a + 2$
$a \leftarrow \frac{e}{2} + 1$
$m \leftarrow 5 \cdot 8$
$x \leftarrow m \cdot 2$
$m \leftarrow m + e$
$e \leftarrow e - 1$
$m \leftarrow m + e$
$e \leftarrow e - 1$
$m \leftarrow m + e$
$e \leftarrow e - 1$
$t \leftarrow x + 3$
Gib aus: m

- Fiktiver Algorithmus ohne Kontext
- Aufgabe: Algorithmus kompakt aufbereiten
- Nutzbarmachung von Strukturelementen

Aufgabe 1: Kompaktes, effizientes Programmieren

$a \leftarrow 3$
$e \leftarrow a + 2$
$a \leftarrow \frac{e}{2} + 1$
$m \leftarrow 5 \cdot 8$
$x \leftarrow m \cdot 2$
$m \leftarrow m + e$
$e \leftarrow e - 1$
$m \leftarrow m + e$
$e \leftarrow e - 1$
$m \leftarrow m + e$
$e \leftarrow e - 1$
$t \leftarrow x + 3$
Gib aus: m

Optimale Lösung

$a \leftarrow 3$
$e \leftarrow a + 2$
$m \leftarrow 5 \cdot 8$
$e \geq 3$
$m \leftarrow m + e$
$e \leftarrow e - 1$
Gib aus: m

- Fiktiver Algorithmus ohne Kontext
- Aufgabe: Algorithmus kompakt aufbereiten
- Nutzbarmachung von Strukturelementen

Kategorisierung des Datenmaterials: Aufgabe 1

- **Kategorie 1:** Optimale Lösungen

Kategorisierung des Datenmaterials: Aufgabe 1

- **Kategorie 1:** Optimale Lösungen
- **Kategorie 2:** Schleife wurde eingearbeitet. Algorithmus enthält für die Ausgabe irrelevante Anweisungen.

Kategorisierung des Datenmaterials: Aufgabe 1

- **Kategorie 1:** Optimale Lösungen
- **Kategorie 2:** Schleife wurde eingearbeitet. Algorithmus enthält für die Ausgabe irrelevante Anweisungen.
- **Kategorie 3:** Schleife wurde eingearbeitet. Ausgabe des Algorithmus unterscheidet sich von der Ausgabe des vorgegebenen Algorithmus.

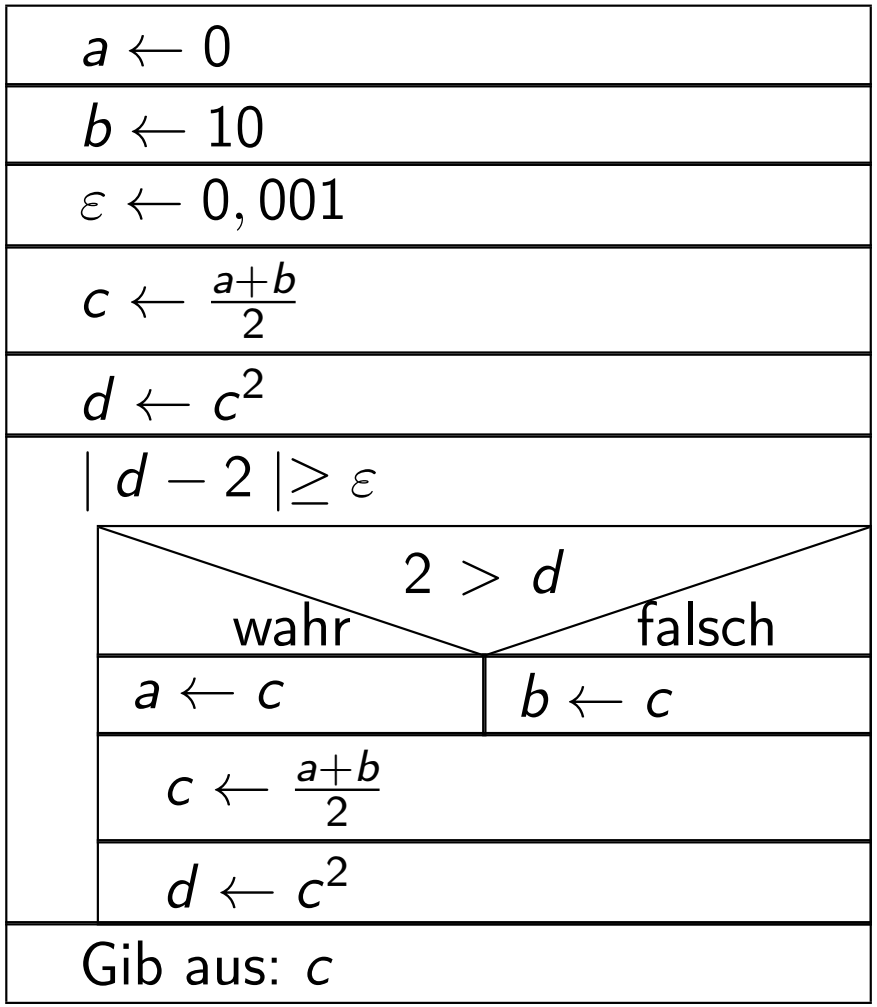
Kategorisierung des Datenmaterials: Aufgabe 1

- **Kategorie 1:** Optimale Lösungen
- **Kategorie 2:** Schleife wurde eingearbeitet. Algorithmus enthält für die Ausgabe irrelevante Anweisungen.
- **Kategorie 3:** Schleife wurde eingearbeitet. Ausgabe des Algorithmus unterscheidet sich von der Ausgabe des vorgegebenen Algorithmus.
- **Kategorie 4:** Schleife wurde erkannt. Lösungsvorschläge wurden nicht erbracht.

Kategorisierung des Datenmaterials: Aufgabe 1

- **Kategorie 1:** Optimale Lösungen
- **Kategorie 2:** Schleife wurde eingearbeitet. Algorithmus enthält für die Ausgabe irrelevante Anweisungen.
- **Kategorie 3:** Schleife wurde eingearbeitet. Ausgabe des Algorithmus unterscheidet sich von der Ausgabe des vorgegebenen Algorithmus.
- **Kategorie 4:** Schleife wurde erkannt. Lösungsvorschläge wurden nicht erbracht.
- **Kategorie 5:** Falscher bzw. diffuser Ansatz

Aufgabe 2: Korrektur logischer Fehler



- Kontext: Algorithmus soll Näherungswert für $\sqrt{2}$ ausgeben
- Aufgabe: Fehlersuche

- Pretest : Fehler in Schleifeneintrittsbedingung
- Posttest: Fehler in der Selektionsbedingung

Kategorisierung des Datenmaterials: Aufgabe 2

- **Kategorie 1:** Fehlerverursachender Baustein wird gefunden; Korrektur gemäß der Aufgabenstellung

Kategorisierung des Datenmaterials: Aufgabe 2

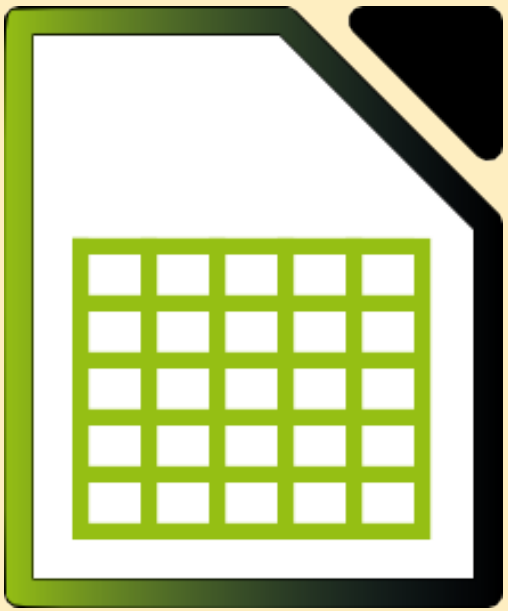
- **Kategorie 1:** Fehlerverursachender Baustein wird gefunden; Korrektur gemäß der Aufgabenstellung
- **Kategorie 2:** Fehlerverursachender Baustein wird gefunden; Korrektur entspricht nicht der Aufgabenstellung

Kategorisierung des Datenmaterials: Aufgabe 2

- **Kategorie 1:** Fehlerverursachender Baustein wird gefunden; Korrektur gemäß der Aufgabenstellung
- **Kategorie 2:** Fehlerverursachender Baustein wird gefunden; Korrektur entspricht nicht der Aufgabenstellung
- **Kategorie 3:** Proband bearbeitet die Aufgabe ohne zielführenden Ansätze

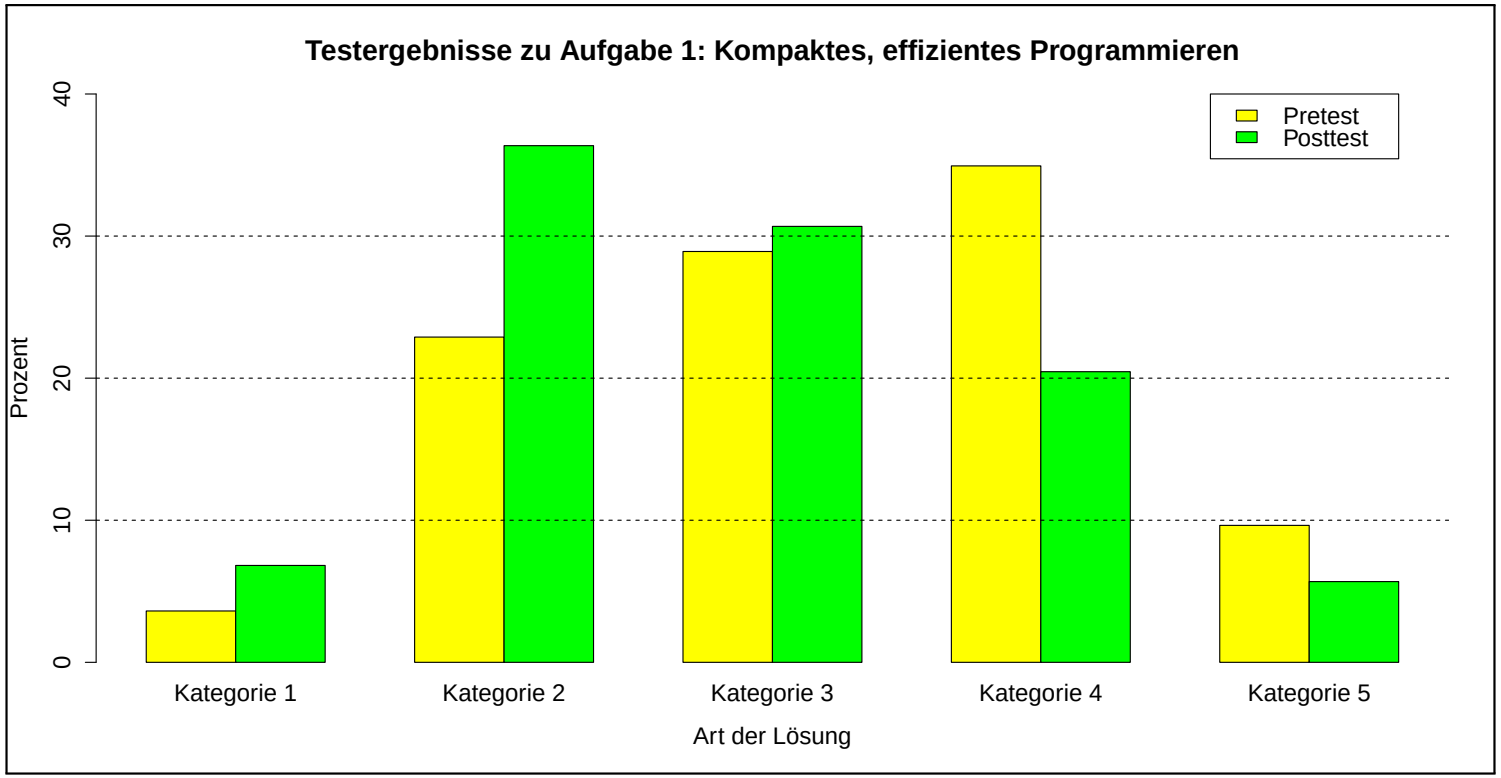
Datenauswertung

Libre Office Calc



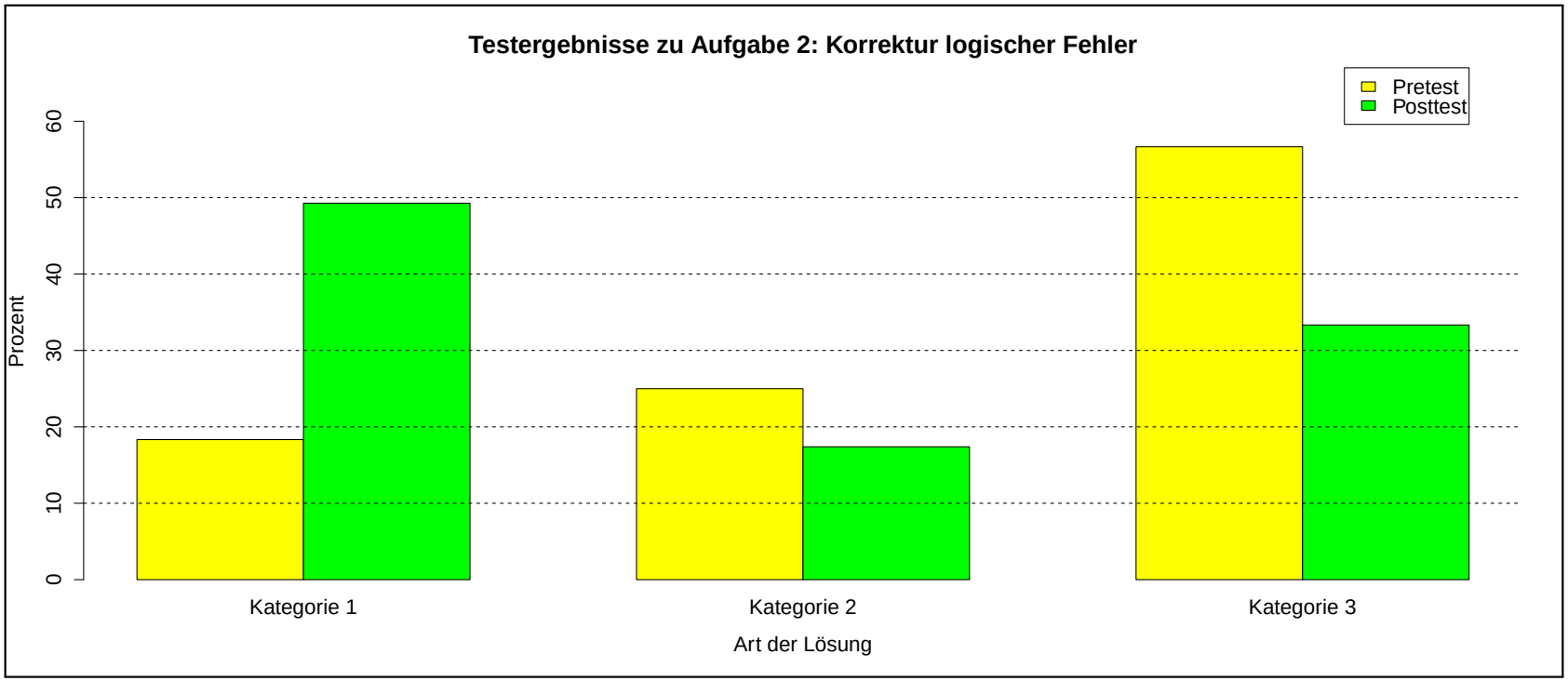
R Studio

Testergebnisse: Programmierungskompetenzen



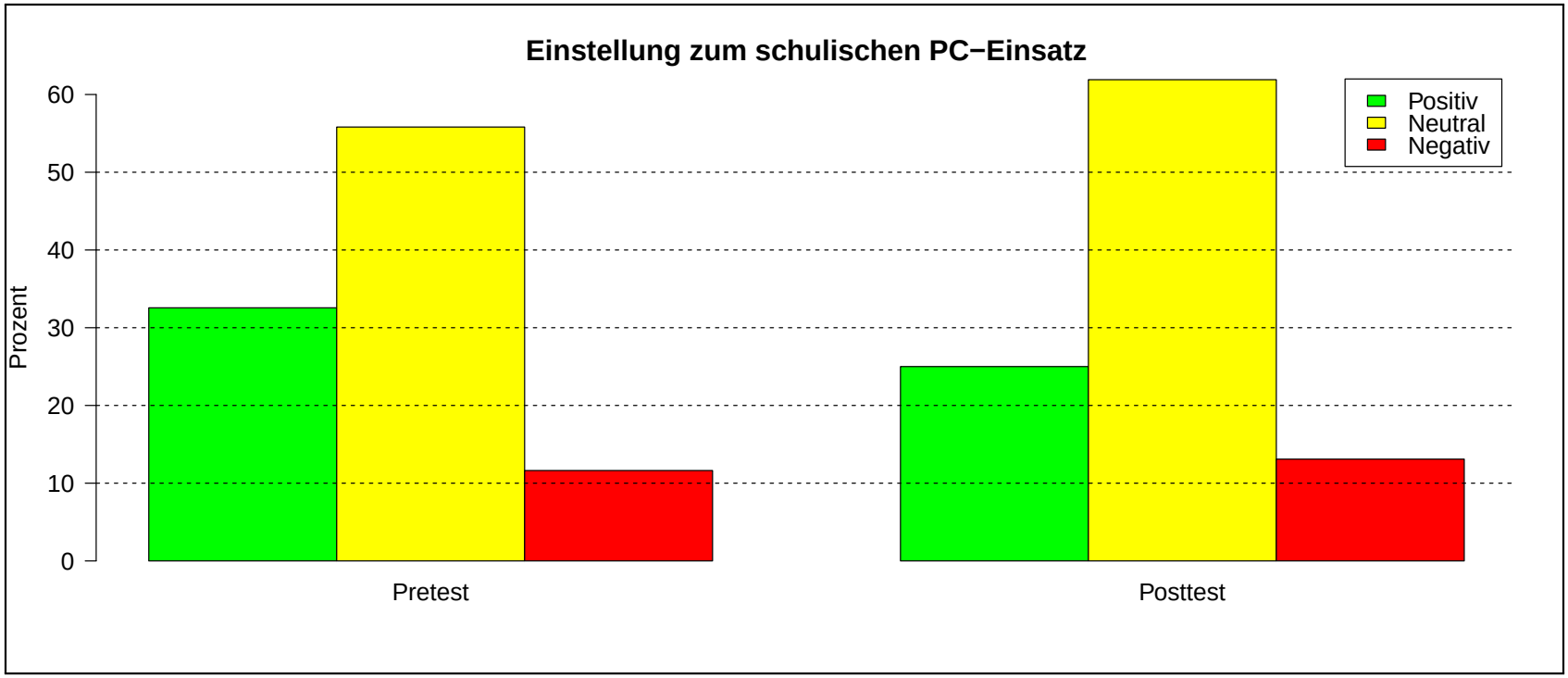
- Wilcox-Test: Signifikanter Kompetenzzuwachs ($p = 0,3\%$)
- Nicht bearbeitet: Pretest: 5,7 % Posttest: 0%

Testergebnisse: Programmierungskompetenzen



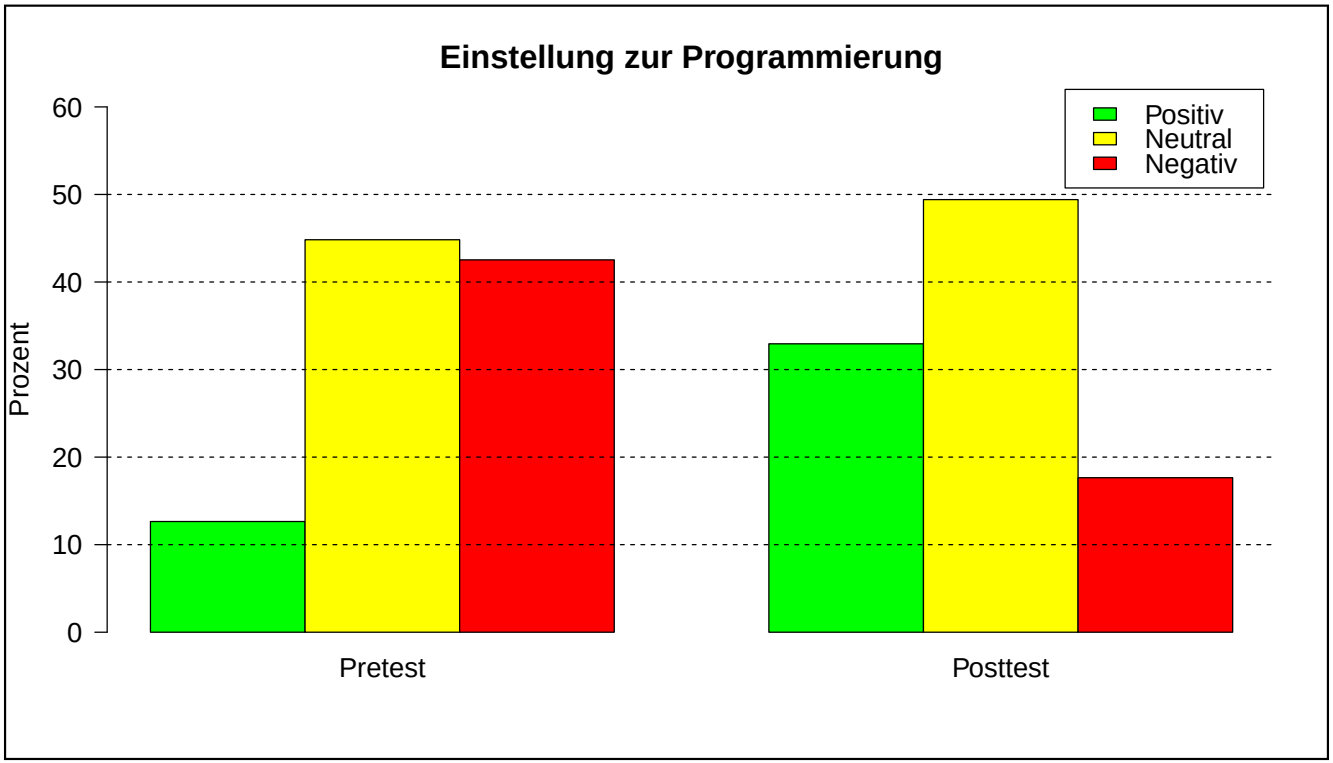
- Wilcox-Test: Signifikanter Kompetenzzuwachs ($p = 0,03\%$)
- Nicht bearbeitet: Pretest: 31,8 % Posttest: 21,6%

Einstellung: PC-Einsatz in der Schule



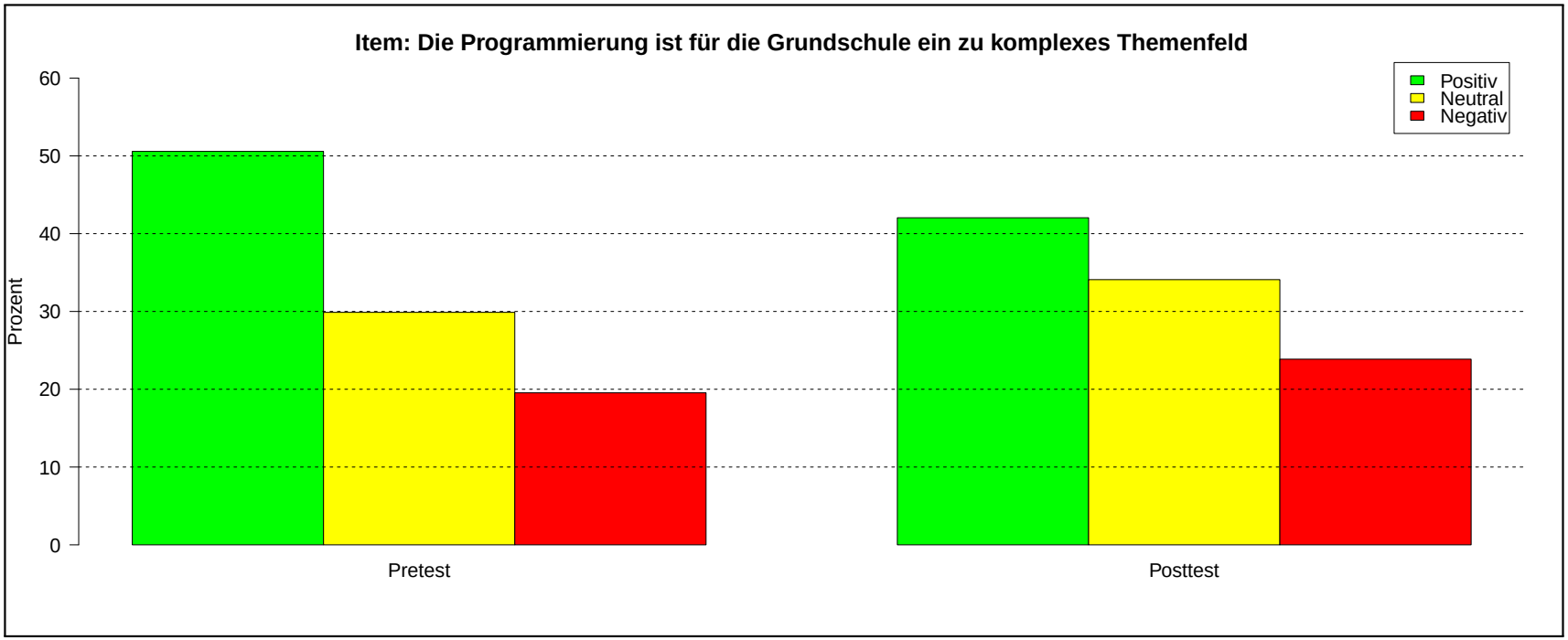
- Beobachtung: Etwas negativere Einstellung im Posttest

Einstellung: Programmierung



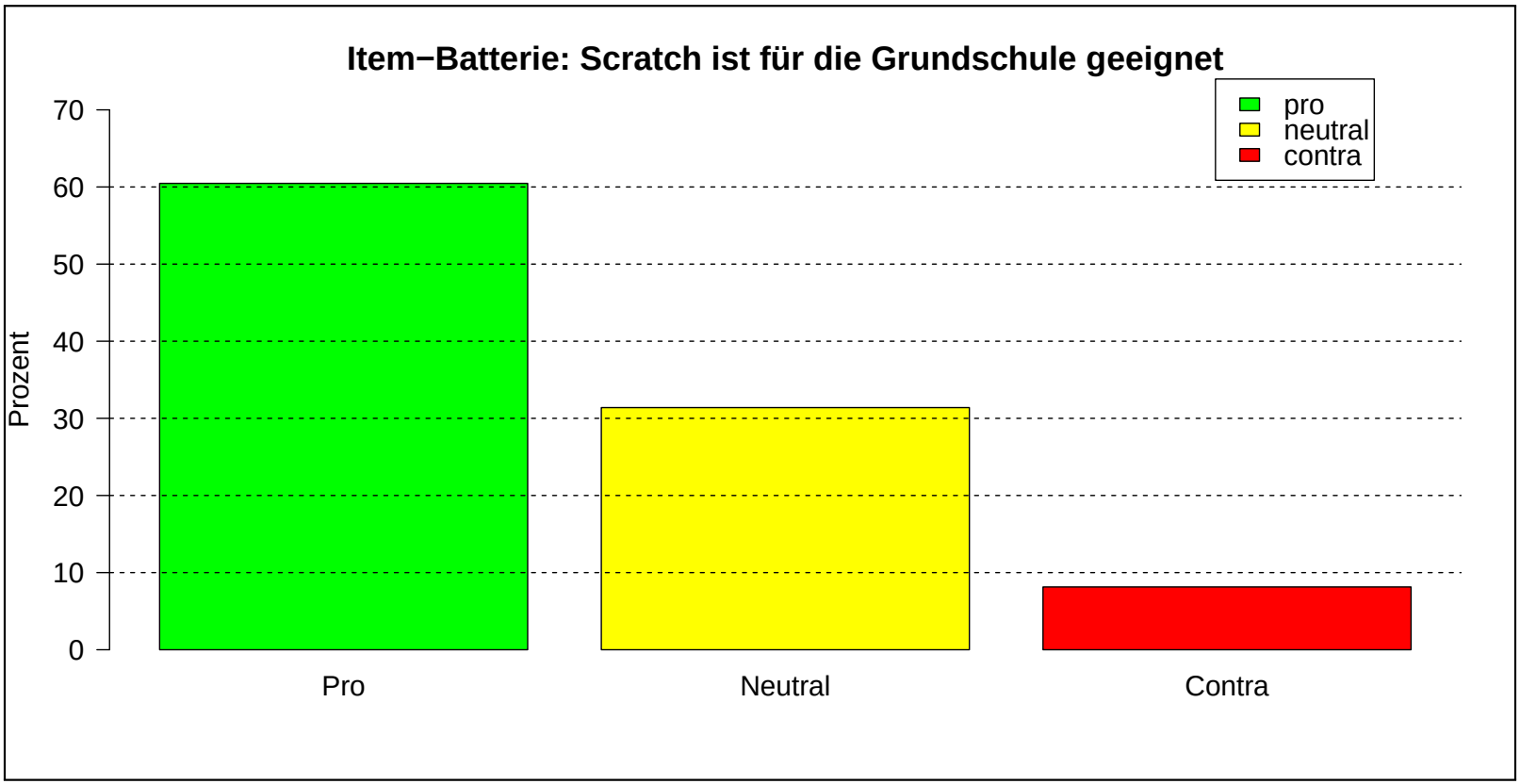
- Einstellungsänderung zugunsten der Programmierung
- Signifikanter Einstellungswandel (Wilcox-Test: $p = 2 \cdot 10^{-5}$)

Einstellung: Programmierung in der Grundschule?



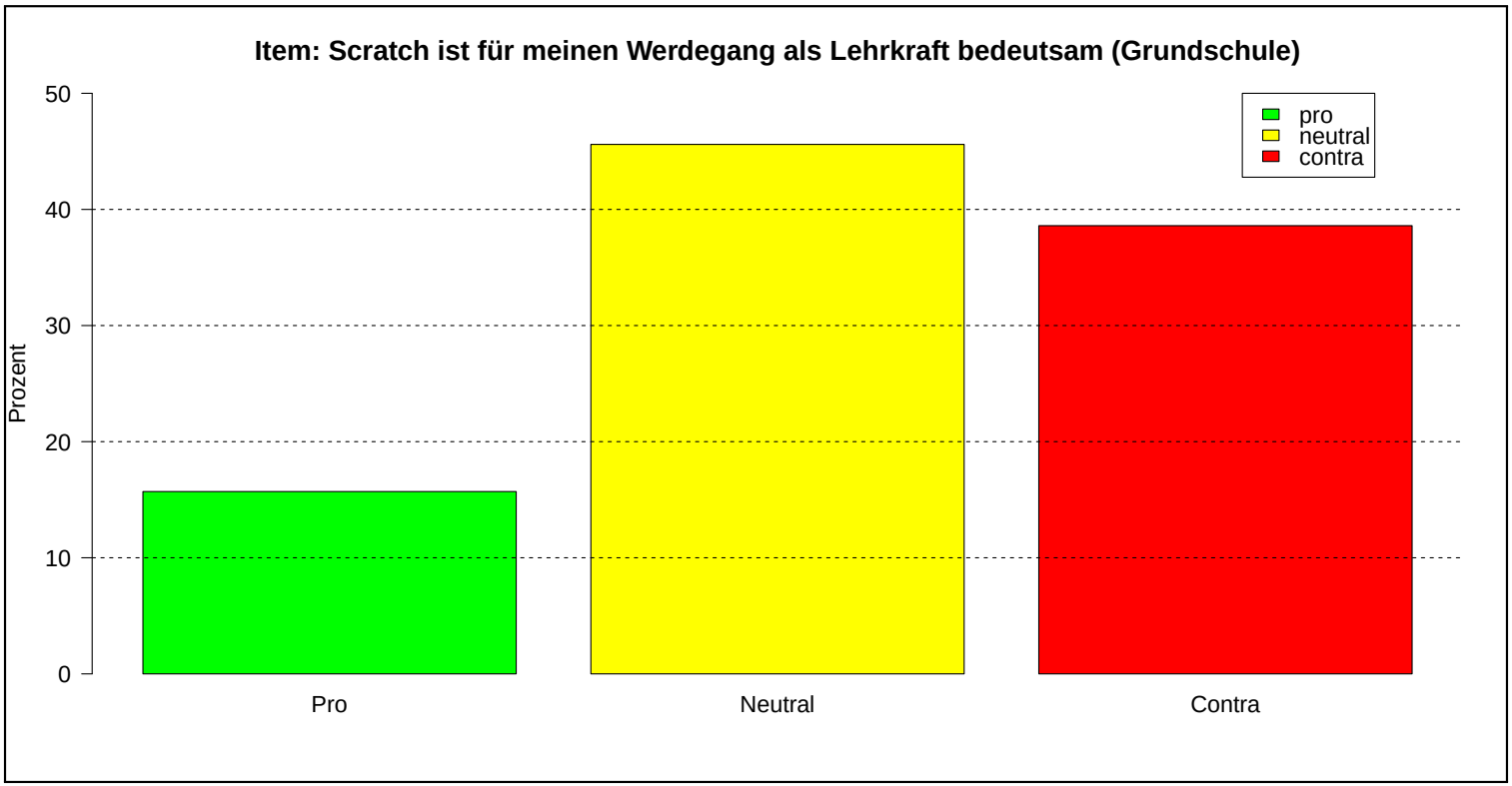
- Meinung: Programmierung ist zu komplex für die Grundschule

Einstellung: Scratch in der Grundschule?



- Meinung: Scratch ist für die Grundschule geeignet

Einstellung: Scratch für meinen Werdegang?



- Meinung: Grundschullehrer halten Scratch nicht für ihren Werdegang bedeutsam.

Inhalt

1. Scratch: Theoretische Grundlagen
   - Motivation und Ziele von Scratch
   - Wesen und Funktionsweise
   - Forschungsstand
2. Empirische Studie
   - Begründung und Ziel des empirischen Vorhabens
   - Forschungsdesign
   - Methoden der Datenerhebung- und Auswertung
   - Darstellung und Interpretation der Daten
3. **Schlussfolgerungen**
   - Erkenntnisse aus der Untersuchung
   - Umgang mit Scratch

Erkenntnisse der Untersuchung

Aneignung von Programmierungskompetenzen mit Scratch

- Deutlicher Kompetenzzuwachs:
  - Kompaktes, effizientes Programmieren
  - Korrektur logischer Fehler

Erkenntnisse der Untersuchung

Aneignung von Programmierungskompetenzen mit Scratch

- Deutlicher Kompetenzzuwachs:
  - Kompaktes, effizientes Programmieren
  - Korrektur logischer Fehler

Einstellung zur Programmierung

- Einstellungswandel zugunsten der Programmierung allgemein
- Aber: Fachlicher Anspruch nicht mit Grundschule vereinbar

Erkenntnisse der Untersuchung

Aneignung von Programmierungskompetenzen mit Scratch

- Deutlicher Kompetenzzuwachs:
  - Kompaktes, effizientes Programmieren
  - Korrektur logischer Fehler

Einstellung zur Programmierung

- Einstellungswandel zugunsten der Programmierung allgemein
- Aber: Fachlicher Anspruch nicht mit Grundschule vereinbar

Einstellung zu Scratch

- Eignung für die Grundschule wird ausgesprochen
- Aber: Studierende sehen keine Bedeutung für ihren weiteren Werdegang

Schlussfolgerungen für den Umgang mit Scratch

Zielgerichteter Einsatz Scratchs in der universitären Lehre

- Vermittlung von Programmierungskompetenzen
- Abbau negativer Haltungen gegenüber der Programmierung
- Scratch als Angebot für die Grundschule

Schlussfolgerungen für den Umgang mit Scratch

Zielgerichteter Einsatz Scratchs in der universitären Lehre

- Vermittlung von Programmierungskompetenzen
- Abbau negativer Haltungen gegenüber der Programmierung
- Scratch als Angebot für die Grundschule

Zu beachten:

- Kritischer Umgang ist unerlässlich
- Ziele des Unterrichts bestimmen den Einsatz einer Programmiersprache
- Alternativen anbieten

Zusammenfassung

1. Scratch eignet sich für die Vermittlung von Programmierungskompetenzen.
2. Scratch baut negative Haltungen gegenüber der Programmierung ab.

Zusammenfassung

1. Scratch eignet sich für die Vermittlung von Programmierungskompetenzen.
2. Scratch baut negative Haltungen gegenüber der Programmierung ab.

Ausblick - Forschung

- Können Programmierungskompetenzen bereits in der Grundschule entwickelt werden?
- Unterstützen Programmierungskompetenzen notwendige mathematische Kompetenzen der Grundschule?

Diskussion

Wie kann eine unterrichtliche Aufbereitung von Programmierungsaspekten in der Grundschulpraxis (mittels Scratch) gelingen?