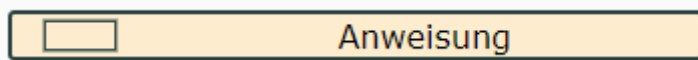
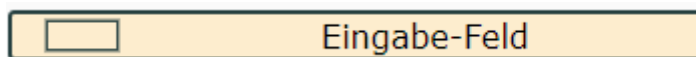


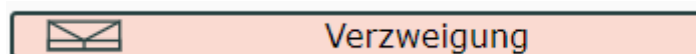
Struktogramme



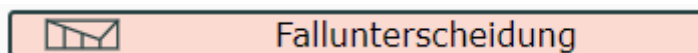
- Wann: Für einfache, einzelne Befehle oder Operationen
- Wie: Rechteckiger Block mit der Anweisung im Inneren
- Beispiel: "x = 5" oder "Drucke 'Hallo Welt'"



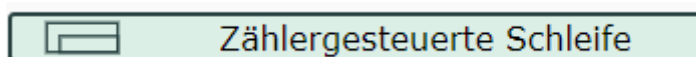
- Wann: Zum Einlesen von Daten von der Benutzeroberfläche oder einer Datei
- Wie: Parallelogramm mit der Eingabebeschreibung
- Beispiel: "Lies Benutzername"



- Wann: Für Ja/Nein-Entscheidungen im Programmablauf
- Wie: Rechteck mit diagonaler Teilung, Bedingung oben, "Ja" links, "Nein" rechts
- Beispiel: "Ist x > 0?"



- Wann: Bei mehreren möglichen Fällen oder Optionen
- Wie: Rechteck mit vertikalen Unterteilungen für jeden Fall
- Beispiel: "switch(farbe)" mit Fällen für Rot, Grün, Blau



- Wann: Für Schleifen mit einer festgelegten Anzahl von Durchläufen
- Wie: Rechteck mit horizontaler Linie oben, Schleifenbedingung darüber
- Beispiel: "Für i von 1 bis 10"



Kopfgesteuerte Schleife

- Wann: Für Schleifen, bei denen die Bedingung am Anfang geprüft wird
- Wie: Rechteck mit horizontaler Linie oben, Schleifenbedingung darüber
- Beispiel: "Solange $x < 100$ "



Fußgesteuerte Schleife

- Wann: Für Schleifen, bei denen die Bedingung am Ende geprüft wird
- Wie: Rechteck mit horizontaler Linie unten, Schleifenbedingung darunter
- Beispiel: "Wiederhole ... bis $x > 100$ "



Funktionsblock

- Wann: Zur Darstellung von Unterprogrammen oder Funktionsaufrufen
- Wie: Rechteck mit doppelten vertikalen Linien an den Seiten
- Beispiel: "Berechne Summe(a, b)"



Try-Catch-Block

- Wann: Zur Fehlerbehandlung und zum Abfangen von Ausnahmen
- Wie: Zwei nebeneinander liegende Rechtecke, links "Try", rechts "Catch"
- Beispiel: "Versuche Datei zu öffnen" | "Fange FileNotFoundException"

Allgemeine Tipps zur Erstellung von Struktogrammen:

1. Beginnen Sie oben und arbeiten Sie sich nach unten.
2. Verwenden Sie klare, präzise Beschreibungen in den Blöcken.
3. Achten Sie auf eine konsistente Formatierung und Größe der Blöcke.
4. Nummerieren Sie die Blöcke bei Bedarf für bessere Übersichtlichkeit.
5. Verwenden Sie Kommentare, um komplexe Teile zu erklären.

Beim Erstellen eines Struktogramms wählen Sie die Elemente basierend auf der Logik und dem Fluss Ihres Algorithmus aus. Beginnen Sie mit der Hauptstruktur und verfeinern Sie dann die Details. Denken Sie daran, dass Struktogramme dazu dienen, den Algorithmus klar und verständlich darzustellen.

Struktogramm erstellen

```
def schokoriegel_automat():
    # Preise der Schokoriegel
    preise = {
        "1": 1.50, # Schokoriegel A
        "2": 2.00, # Schokoriegel B
        "3": 2.50 # Schokoriegel C
    }

    # Begrüßung und Auswahlmenü
    print("Willkommen beim Schokoriegel-Automaten!")
    print("Bitte wählen Sie eine Sorte:")
    print("1 - Schokoriegel A (1.50€)")
    print("2 - Schokoriegel B (2.00€)")
    print("3 - Schokoriegel C (2.50€)")

    # Benutzereingabe für Schokoriegel-Auswahl
    auswahl = input("Ihre Auswahl: ")

    if auswahl in preise:
        preis = preise[auswahl]
        print(f"Preis für Schokoriegel {auswahl}: {preis:.2f}€")

        # Benutzereingabe für Geldeinwurf
        geld = float(input("Bitte werfen Sie das Geld ein: "))

        if geld >= preis:
            wechselgeld = geld - preis
            print(f"Schokoriegel {auswahl} wird ausgegeben.")
            if wechselgeld > 0:
                print(f"Ihr Wechselgeld beträgt: {wechselgeld:.2f}€")
        else:
            fehlbetrag = preis - geld
            print(f"Nicht genug Geld eingeworfen. Es fehlen noch {fehlbetrag:.2f} €.")
    else:
        print("Ungültige Auswahl. Bitte versuchen Sie es erneut.")

    print("Vielen Dank, dass Sie unseren Automaten benutzt haben!")

# Hauptprogramm starten
schokoriegel_automat()
```

1. Identifiziere die Hauptschritte des Programms:

Begrüßung und Anzeige des Auswahlmenüs

```
# Begrüßung und Auswahlmenü
print("Willkommen beim Schokoriegel-Automaten!")
print("Bitte wählen Sie eine Sorte:")
print("1 - Schokoriegel A (1.50€)")
print("2 - Schokoriegel B (2.00€)")
print("3 - Schokoriegel C (2.50€)")
```

Entgegennahme der Benutzerauswahl

```
# Benutzereingabe für Schokoriegel-Auswahl
auswahl = input("Ihre Auswahl: ")
```

Prüfung der Benutzerauswahl

```
if auswahl in preise:
```

Berechnung des Preises

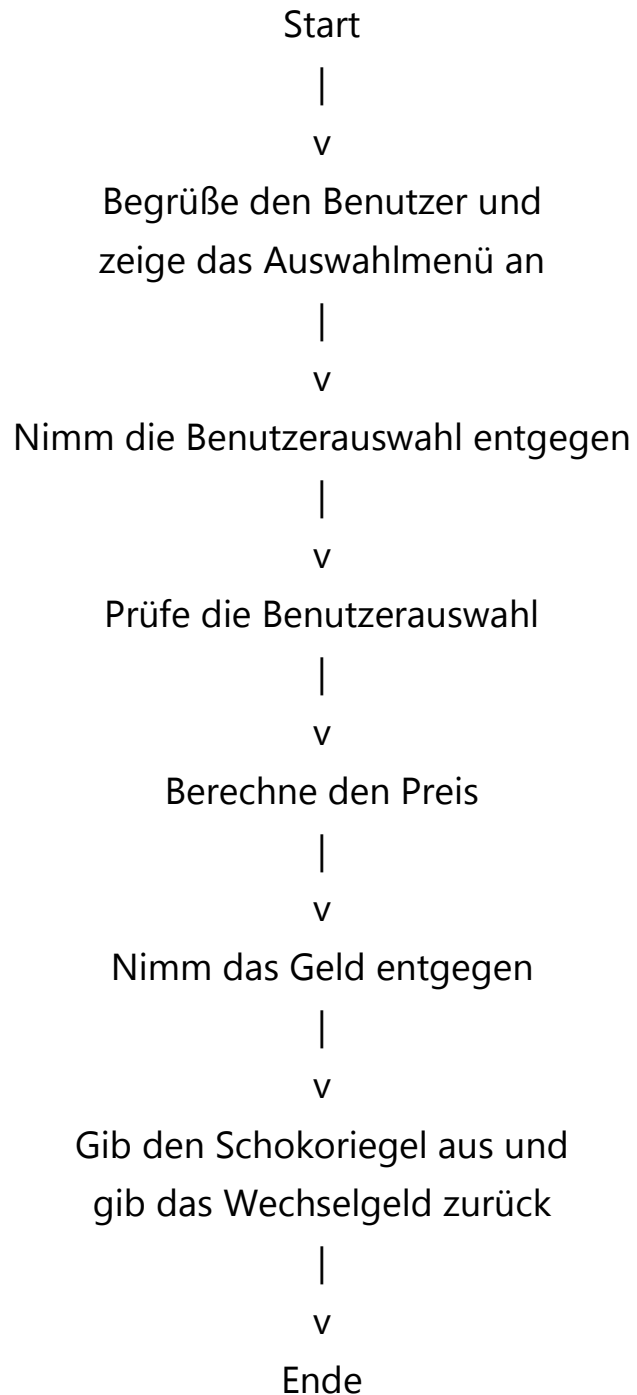
```
# Preise der Schokoriegel
preise = {
    "1": 1.50, # Schokoriegel A
    "2": 2.00, # Schokoriegel B
    "3": 2.50 # Schokoriegel C
}
```

Entgegennahme des Geldes, Ausgabe des Schokoriegels und Rückgabe des Wechselgeldes

```
# Benutzereingabe für Geldeinwurf
geld = float(input("Bitte werfen Sie das Geld ein: "))











if geld >= preis:
    wechselgeld = geld - preis
    print(f"Schokoriegel {auswahl} wird ausgegeben.")
    if wechselgeld > 0:
        print(f"Ihr Wechselgeld beträgt: {wechselgeld:.2f}€")
else:
    fehlbetrag = preis - geld
    print(f"Nicht genug Geld eingeworfen. Es fehlen noch {fehlbetrag:.2f}€")
```

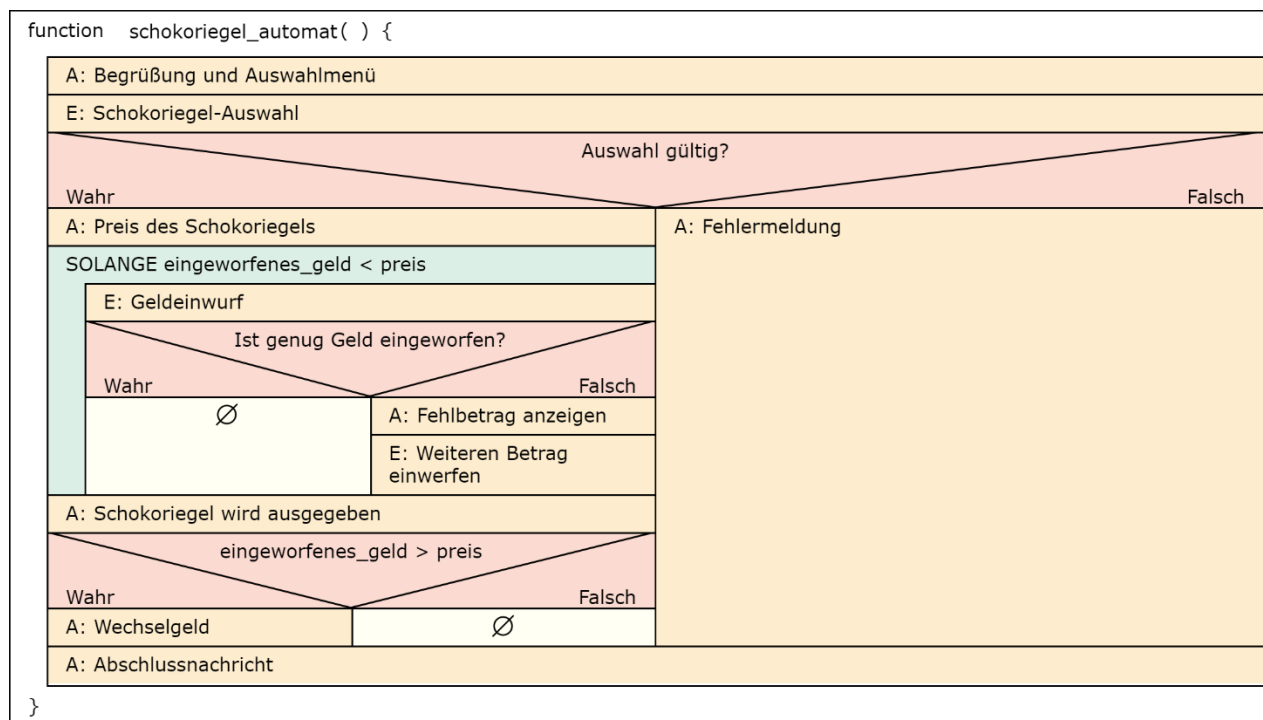
2. Zeichne das Grundgerüst des Struktogramms:



3. Detailliere die einzelnen Schritte und zeichne das Struktogramm:

Überlege welche Elemente du benötigst:

 Eingabe-Feld	 Ausgabe-Feld	 Anweisung
 Zählergesteuerte Schleife	 Kopfgesteuerte Schleife	 Fußgesteuerte Schleife
 Verzweigung	 Fallunterscheidung	 Try-Catch-Block
	 Funktionsblock	



Verwendetes Tool: <https://struktog.openpatch.org/>

4. Vervollständige das Struktogramm:

Überprüfe, ob alle Schritte des Programms im Struktogramm abgebildet sind. Achte auf eine übersichtliche und gut strukturierte Darstellung.