Swtpp-Gedächtsprotokd 1a 17.01.25

1. Multiple Choice

a) Zähle die grundlegenden Aktivitäten eines Softwareentwicklungsprozesses auf.

b) Welche Art von Prozess ist ein Wasserfallmodell?

c) Was ist ein Merkmal des Agilen Entwicklungsprozesses?

d) Wann gilt ein Projekt in Spiralentwicklung als gescheitert?

e) Welches der Folgenden sind Scrum-Artefakte?

f) Was ist die Aufgabe eines Projekt Owners?

2. Testen und Codequalität

a) Gegeben ist die Funktion BerechneWert(int x), welche einen Wert durch die Eingabe x berechnet. Stelle einen Kontrollflussgraphen nach der in der Vorlesung aufgestellten Vorgaben auf. Zeilen dürfen zusammengefasst werden, wenn die Zeilen aus dem Namen des Knotens hervorgehen.

```
berecheWert( int x ) {
zaehler = 0
ergebnis = 0

while (zaehler < 3) {
    if ( x > 2 &&
        zaehler % 2 == 0) {
        ergebnis += x * zaehler
    } else {
        ergebnis += zaehler
    }

    if ( ergebnis > 10 ) {
        return -1
    }

    zaehler++
}

return ergebnis

}
```

b) Gebe für x Werte an, um minimale Zweigabdeckung mit möglichst wenig Werten zu erreichen.

c) Gebe weitere Testfälle an, um die minimale Mehrfach-Bedingungsabdeckung zu erreichen.

d) Berechne die Komplexität der Funktion nach McGabe.

3. Prolog

Gegeben ist eine Datenbasis in Prolog.
lebtIn(X,Y) gibt an, in welchen Gebiet Y das Lebewesen X lebt.
bestehtAus(X,Y) gibt an, welche Nahrung Y das Lebewesen X liefert.
braucht(X,Y) gibt an, welche Nahrung Y das Lebenwesen X zu leben braucht.

```
lebtIn(loewe,savanne).
lebtIn(nashorn,savanne).
lebtIn(gras,savanne).
lebtIn(eisbaer,antarktis).
lebtIn(pinguin,antarktis).
lebtIn(wal,antarktis).
lebtIn(eisfisch,antarktis).
lebtIn(alge,antarktis).
lebtIn(tiger,regenwald).
lebtIn(orangutan,regenwald).
lebtIn(banane,regenwald).
```

```
bestehtAus(loewe,fleisch).
bestehtAus(nashorn,fleisch).
bestehtAus(gras,pflanze).
bestehtAus(eisbaer,fleisch).
bestehtAus(pinguin,fleisch).
bestehtAus(wal,fleisch).
bestehtAus(eisfisch,fisch).
bestehtAus(alge,pflanze).
bestehtAus(tiger,fleisch).
bestehtAus(orangutan,fleisch).
bestehtAus(banane,pflanze).
```

```
braucht(loewe,fleisch).
braucht(nashorn,pflanze).
braucht(gras,licht).
braucht(eisbaer,fleisch).
braucht(pinguin,fisch).
braucht(wal,pflanze).
braucht(eisfisch,pflanze).
braucht(alge,licht).
braucht(tiger,fleisch).
braucht(orangutan,pflanze).
braucht(banane,licht).
```

a) Gebe eine Query an, die auf der Variable X zurückgibt, wo der Loewe lebt.
?-

b) Gebe eine Query an, die auf der Liste L zurückgibt, welche Lebewesen in der Savanne leben.
?-

c) Gebe ein Prädikat **gleicherLebensraum(X,Y)** an, welches für X und Y überprüft, ob sie den gleichen Lebensraum haben.

d) Gebe ein Prädikat **gleicheNahrung(L,X)** an, die für eine Liste L von Lebewesen überprüft, ob diese die Nahrung X brauchen.

e) Gebe ein Prädikat **nahrungskette(X,L)** an, welches für ein Lebewesen X mögliche Listen einer Nahrungskette L zurückgibt. Dabei muss am Anfang der Liste ein Lebewesen stehen, welches Fleisch frisst. Zyklen und Wiederholungen sollen vermieden werden.

f) Gebe ein Prädikat **fischOderFleisch(Erg,L)** an, welches zählt, wie viele Lebewesen in der Liste L aus Fleisch oder Fisch bestehen und die Summe in Erg zurückgibt.

Swtpp-Gedächtsprotokd 1a 17.01.25

1. Multiple Choice

a) Zähle die grundlegenden Aktivitäten eines Softwareentwicklungsprozesses auf.

-Spezifikation, Entwicklung, Validierung, Evolution.

b) Welche Art von Prozess ist ein Wasserfallmodell?

-sequenzieller Entwicklungsprozess

c) Was ist ein Merkmal des Agilen Entwicklungsprozesses?

-Änderungen sind möglich.

d) Wann gilt ein Projekt in Spiralentwicklung als gescheitert?

-Risiken wurden nicht beseitigt.

e) Welches der Folgenden sind Scrum-Artefakte?

-Produkt Bachlog & Sprint Bachlog

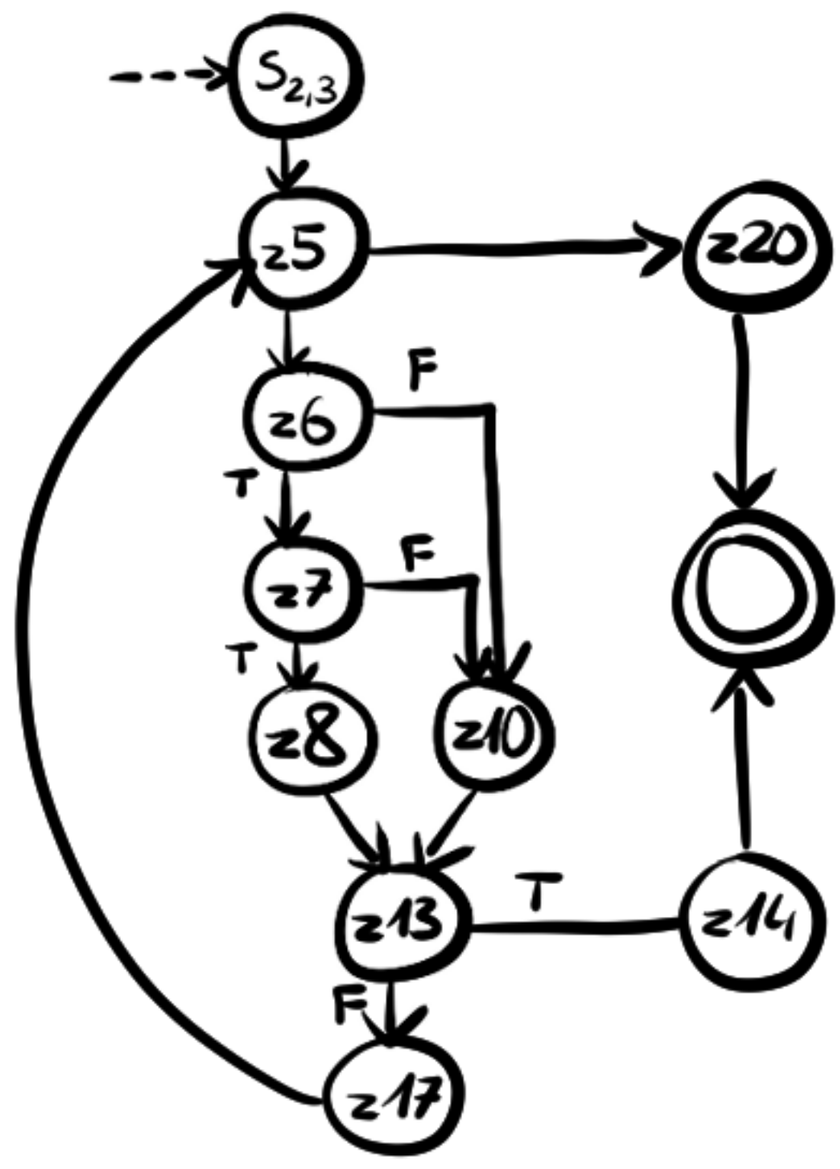
f) Was ist die Aufgabe eines Projekt Owners?

-Produktwertmaximierung, Pflegen Product Bachlog.

2. Testen und Codequalität

a) Gegeben ist die Funktion BerechneWert(int x), welche einen Wert durch die Eingabe x berechnet. Stelle einen Kontrollflussgraphen nach der in der Vorlesung aufgestellten Vorgaben auf. Zeilen dürfen zusammengefasst werden, wenn die Zeilen aus dem Namen des Knotens hervorgehen.

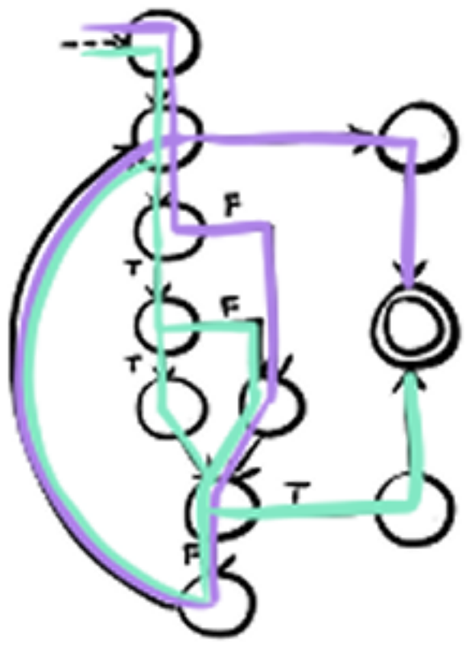
```
   berecheWert( int x ) {
2  zaehler = 0
3  ergebnis = 0

5  while (zaehler < 3) {
6      if ( x > 2 &&
7          zaehler % 2 == 0) {
8          ergebnis += x * zaehler
       } else {
10         ergebnis += zaehler
       }

13     if ( ergebnis > 10 ) {
14         return -1
       }

17     zaehler++
   }

20 return ergebnis

   }
```



Swtpp-Gedächtsprotokd 1a 17.01.25

b) Gebe für x Werte an, um minimale Zweigabdeckung mit möglichst wenig Werten zu erreichen.

x=0 ; x=5



c) Gebe weitere Testfälle an, um die minimale Mehrfach-Bedingungsabdeckung zu erreichen.

x=3

d) Berechne die Komplexität der Funktion nach McGabe.

v(G) = e - n + 2p = 14 - 11 + 2 = 5

3. Prolog

Gegeben ist eine Datenbasis in Prolog.
lebtIn(X,Y) gibt an, in welchen Gebiet Y das Lebewesen X lebt.
bestehtAus(X,Y) gibt an, welche Nahrung Y das Lebewesen X liefert.
braucht(X,Y) gibt an, welche Nahrung Y das Lebenwesen X zu leben braucht.

```
lebtIn(loewe,savanne).
lebtIn(nashorn,savanne).
lebtIn(gras,savanne).
lebtIn(eisbaer,antarktis).
lebtIn(pinguin,antarktis).
lebtIn(wal,antarktis).
lebtIn(eisfisch,antarktis).
lebtIn(alge,antarktis).
lebtIn(tiger,regenwald).
lebtIn(orangutan,regenwald).
lebtIn(banane,regenwald).
```

```
bestehtAus(loewe,fleisch).
bestehtAus(nashorn,fleisch).
bestehtAus(gras,pflanze).
bestehtAus(eisbaer,fleisch).
bestehtAus(pinguin,fleisch).
bestehtAus(wal,fleisch).
bestehtAus(eisfisch,fisch).
bestehtAus(alge,pflanze).
bestehtAus(tiger,fleisch).
bestehtAus(orangutan,fleisch).
bestehtAus(banane,pflanze).
```

```
braucht(loewe,fleisch).
braucht(nashorn,pflanze).
braucht(gras,licht).
braucht(eisbaer,fleisch).
braucht(pinguin,fisch).
braucht(wal,pflanze).
braucht(eisfisch,pflanze).
braucht(alge,licht).
braucht(tiger,fleisch).
braucht(orangutan,pflanze).
braucht(banane,licht).
```

a) Gebe eine Query an, die auf der Variable X zurückgibt, wo der Loewe lebt.

?- lebtIn(loewe, X).

b) Gebe eine Query an, die auf der Liste L zurückgibt, welche Lebewesen in der Savanne leben.

?- FindAll(X, lebtIn(X, savanne), L).

Swtpp-Gedächtsprotokd 1a 17.01.25

c) Gebe ein Prädikat **gleicherLebensraum(X,Y)** an, welches für X und Y überprüft, ob sie den gleichen Lebensraum haben.

```
gleicherLebensraum(X,Y):- lebtIn(X,Z), lebtIn(Y,Z).
```

d) Gebe ein Prädikat **gleicheNahrung(L,X)** an, die für eine Liste L von Lebewesen überprüft, ob diese die Nahrung X brauchen.

```
gleicheNahrung([],_).
gleicheNahrung([H|T],X):- braucht(H,X), gleicheNahrung(T,X).
```

e) Gebe ein Prädikat **nahrungskette(X,L)** an, welches für ein Lebewesen X mögliche Listen einer Nahrungskette L zurückgibt. Dabei muss am Anfang der Liste ein Lebewesen stehen, welches Fleisch frisst. Zyklen und Wiederholungen sollen vermieden werden.

```
nahrungskette(X,[X]):- braucht(X,Fleisch).
nahrungskette(X,L):- braucht(Y,Z),
                     besteht Aus(X,Z),
                     gleicherLebensraum(X,Y),
                     \+gleicheNahrung([X,Y],_)
                     append(Erg,X,L),
                     nahrungskette(Y,Erg).
```

f) Gebe ein Prädikat **fischOderFleisch(Erg,L)** an, welches zählt, wie viele Lebewesen in der Liste L aus Fleisch oder Fisch bestehen und die Summe in Erg zurückgibt.

```
FischOderFleisch(0,[]).
FischOderFleisch(Erg,[H|T]):-
        (besteht Aus(H,Fisch);besteht Aus(H,Fleisch)),
        FischOderFleisch(Erg1,T),
        Erg is Erg1+1.
FischOderFleisch(Erg,[H|T]):-
        \+(besteht Aus(H,Fisch);besteht Aus(H,Fleisch)),
        FischOderFleisch(Erg,T).
```