Leibniz Universität Hannover

Programmieren 1

Auditorium Exercise 9

Human-Computer Interaction Group

Jan Feuchter
programmieren1@hci.uni-hannover.de

Organisatorisches: Online Sprechstunden

- Ähnlich zur LernLounge
  - Individuelle Hilfe bei Problemen mit Übungsaufgaben
  - Aufzeigen von Ansätzen
  - Keine Herausgabe von Lösungen

- Montags 18:00-20:00
- Mittwochs 17:00-19:00
- Donnerstags 17:00-19:00

Organisatorisches: Weihnachtspause

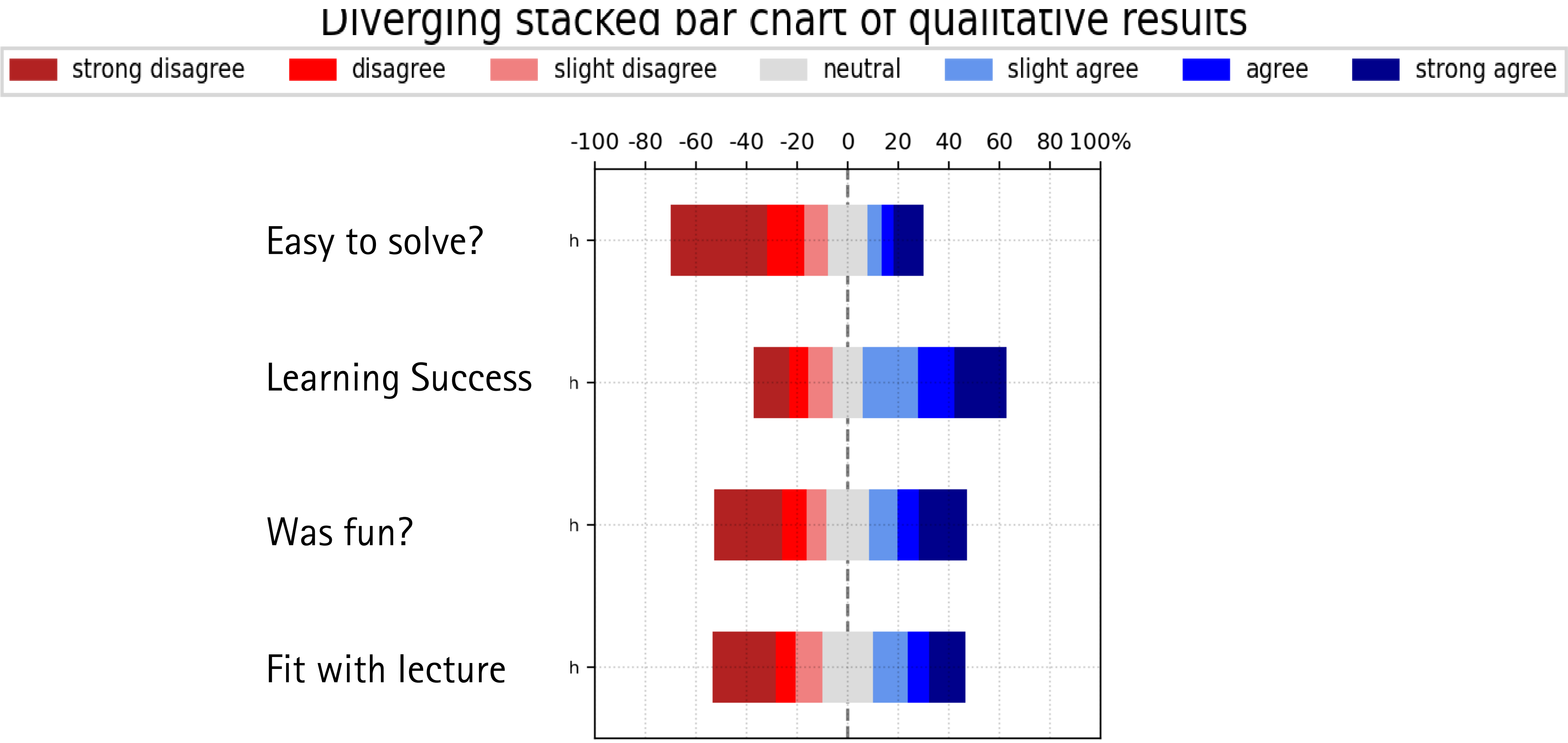
- Nächste Woche (12.12. – 16.12.) normaler Betrieb

- Danach (19.12. – 22.12.) Online-Lehre
    - Übungen per BBB im Stud.IP (Tab „Meetings“)
    - Keine VL + Hörsaalübung am 23.12.

Reversi-Challenge

ASSIGNMENT 8

Feedback – Assignment 8



Assignment 8

INDENTATION

Manually Iterating over a String

```
int indentation(char * s) {
    require_not_null(s);
    int spaces = 0;
    for(char c = *s; c != '\0'; c = *++s) {
        if (c == '\t') return -1;
        if (c != ' ') return spaces;

        // assert(c == ' ')
        spaces++;
    }
    return spaces;
}
```

Manually Iterating over a String (2)

```
for(char c = *s; c != '\0'; c = *++s) {…}
```

- char c
  - Not used to represent the position, but the current character
  - In fact, the variable char * s is also modified and points to the current position
- *s: Initially set c to the string's first character
  - Dereferencing the string (char *) yields the first character
- *++s: Move through the string via pointer arithmetic
  - ++s: Perform s = s + 1 and return s + 1
  - s therefore now points to the next character
  - s is dereferenced (*s) and the next character is stored in c
- c != ‘\0’: This loop iterates until c contains the ending null byte

Pointer Arithmetic

- Given: char * c = (...)
  - The statement c + 1 increases the address stored in c by 1 byte
- Given: int * i = (...)
  - The statement i + 1 increases the address stored in i by 4 bytes
  - (On the architectures we are using anyway)

- Because: Pointer Arithmetic takes into account the width of the type

Trimming the Beginning of a String

- `return s + i`
- Return a pointer to somewhere of the middle of the string
- Further processing of this pointer will treat this position as the string's beginning
  - i.e. "skip" the first characters

```
char * left_trim(char * s) {
    require_not_null(s);

    int i = indentation(s);
    if(i >= 0) {
        return s + i;
    } else {
        return "";
    }
}
```

Extracting a C-Style Comment

```c
char * extract_comment(char * s) {
    require_not_null(s);

    char previous_c = '\0';
    for(char c = *s; c != '\0'; c = *++s) {
        if(previous_c == '/' && c == '/') {
            return left_trim(++s);
        }
        previous_c = c;
    }
    return "";
}
```

Questions?

Assignment 9

- Already available on StudIP

- We will have a brief look inside now