

1 Cíle mé ročníkové práce

Popis cíle mé práce. Silde má block s cílem práce.

Koncept knihovny je podobný geogebře. Cílem je zmenšit zátěž a umožnit tak více výpočtů.

Prezentace se soustředí převážně na obecné koncepty a zásady vývoje knihoven.

Konkrétní implementace je dlouhá a nezajímavá, v případě zájmu bude prostor na dotazy.

2 Programovací jazyk TypeScript

Popis jazyka TypeScript. Slide je výčet vlastností jazyka - 3 Body

Rozšíření JavaScriptu. Přidává statické typy. Transpilovaný jazyk - překládá se do JavaScriptu.

Ukázka funkce filter v JavaScriptu

Ukázka funkce filter v TypeScriptu

Ukázka funkce filter v TypeScriptu s využitím moderních funkcí jazyka

Tyto funkce nelze použít v JavaScriptu, protože nemusí být podporovány ve všech prohlížečích.

3 Co jsou knihovny

Definice softwarové knihovny

Knihovna je soubor funkcí, které mohou být použity v jiných programech.

Důležité vlastnosti knihoven - 5 Bodů

- Znovupoužitelnost
- Stabilita
- Dokumentace
- Testování
- Rozšiřitelnost

Vlastnosti jsou důležité i pro normální programy, nicméně u menších (osobních) projektů je to často zanedbáváno (oprávněně).

4 Vývoj knihoven

Proč jsem zvolil vývoj knihovny

Vývoj knihoven je stejný jako vývoj velkých aplikací. Knihovnu ale můžu udělat malou a sám a ukázat na tom koncepty vývoje velkých softwarových projektů.

4.1 Verzování

Verzování kódu a co nám umožňuje

- Uchovávat historii změn
- Spolupráce
- V případě chyb se vrátit k funkční verzi

Zároveň nám umožňuje hladce pracovat na více zařízeních a na více aspektech projektu zároveň.

Git a GitHub, výčet důvodů, proč jsem zvolil právě Git s GitHubem

Zvolil jsem Git jako verzovací systém a GitHub jako platformu pro ukládání kódu.

Zároveň tedy nemusím řešit ukládání a zálohování kódu, přenos mezi zařízeními.

- Nejrozšířenější
- GitHub nabízí výhodné open-source plány
- Snadná vyhledatelnost + stránka projektu

Ukázka stránky projektu na GitHubu

Můžete procházet soubory, historii změn i se podílet na vývoji.

Ukázka stránky s Issues Kdokoli může nahlásit chybu nebo požadavek na novou funkci.

Hodí se i pro mě, abych měl přehled o tom, co je potřeba udělat.

Ukázka stránky s Commity Commit je jedna ucelená změna v kódu. Při psaní úplně nového projektu je ze začátku náročné rozdělit změny na menší části.

Verzování vydání - definice a číslování verzí

Vydání je označení určité verze kódu.

Semantic Versioning je relativně standardní způsob číslování verzí. Ne každý se jím řídí, ale je bezpochybně nejrozšířenější standard.

Ukázka releases na GitHubu

4.2 Správa závislostí a publikace

Definice správce balíčků - pozor dodatek 1 Bod

Správce balíčků je nástroj, který nám umožňuje stahovat a spravovat závislosti.

V našem případě se stará i o publikaci naší knihovny.

Ukázka knihovny na npm

Vybral jsem si npm registr, protože je nejrozšířenější a nejvíce podporovaný.

I *lepší* správci balíčků, jako třeba pnpm, používají npm registr.

4.3 Automatizované testování

Block s důvody automatizovaného testování Šetří nám práci s testováním. Ve finále jsou spolehlivější. Můžeme všechny testy znovu spustit při každé změně a zaručíme tak, že jsme něco nezničili.

Testy vlastně specifikují požadavky pro naši knihovnu. (Test Driven Development)

Ukázka spuštěných testů

Běží lokálně. Neefektivní proces, i jen pár testů na 24jádrovém procesoru trvá několik vteřin.

Zaručuje nám ale absolutně izolované prostředí.

Ukázka GitHub Actions včetně testů

GitHub Actions je nástroj, který nám umožňuje spouštět různé akce při git událostech.

Já mám nastaveny akce na push a pull request (tedy pokud se změní kód, nebo se jeho změna navrhne).

GitHub Actions umí spouštět i jiné akce, než jen testy.

Ukázka testů ve 3 různých prostředích

Testy běží na 3 různých prostředích, abychom zaručili, že knihovna bude fungovat všude.

Můžeme si to dovolit, GitHub má zdarma dostupné prostředí pro Actions pro veřejné repozitáře.

4.4 Programátorská a uživatelská dokumentace

Block o uživatelské dokumentaci

Jednoduchá srozumitelná dokumentace, měla by popisovat, jak všechny části knihovny použít.

U neknihovnových projektů popíše například instalaci a ovládání.

U knihovny může být cílena na lehce pokročilé uživatele.

Block o TypeDocu

Typedoc je nástroj, který nám umožňuje generovat dokumentaci z kódu a komentářů v něm.

Příklad JSDoc komentáře funkce fibonacci

Ukázka JSDoc komentáře na metodě normalize přímo knihovny

Toto je přímo příklad z mého kódu.

Ukázka, jak se metoda projeví v IntelliSense

Toto je zařízeno primárně TypeScriptem.

Ukázka dokumentace detailu v IntelliSense Toto se automaticky zobrazí, pokud funkci chceme použít a vyplnit její argumenty.

Ukázka dokumentace vygenerované TypeDocem

Toto je ve webové dokumentaci knihovny.

Block o programátorské dokumentaci - 1 Bod

Popisuje jak je rozložen kód v knihovně, proč se co kde děje.

Pomáhá jiným vývojářům, nebo nám samotným, pokud se k projektu vrátíme po delší době.

Pomáhá i při vývoji, protože je náročnější udělat špatné rozhodnutí, když ho nejdříve musíme popsat a vysvětlit, proč ho chceme udělat.

Block s GitHub Wiki

Ukázka GitHub Wiki

GitHub Wiki je členěná na stránky, které můžeme propojovat.

Psána v Markdownu, stejně jako README.

5 Příklady využití knihovny

Sestavení minimalistického geometrického modelu - 6 Bodů

Zdroje - 1 Bod

Zdroje na prameny necitované v textu ročníkové práce. Hlavním zdrojem je práce jako taková.

Odkazy

Odkazy na součásti projektu, všechno veřejné a dostupné.