



MATURITNÍ ZKOUŠKY Z IVT

školní rok: 2019 - 2020

studium: čtyřleté

forma: písemná maturitní práce a její obhajoba

Didaktický záměr nové podoby maturitní zkoušky:

Cíle:

- Studenti se učí přemýšlet jako počítačové vědci a aplikovat algoritmické strategie při řešení problémů.
- Studenti jsou schopni vytvořit funkční a praktické počítačové programy doplněné o dokumentaci, popis procesu tvorby a hloubkové vysvětlení algoritmů.

Oba cíle jsou stanoveny s ohledem na využitelnost při následném studiu na vysoké škole.

Záměrem je, aby maturitní práce byla pro studenty více tvůrčí, zajímavější a efektivnější z hlediska naplňování kognitivních cílů oproti dosavadní ústní zkoušce.

Práce se bude rozvíjet postupně. Studenti budou odevzdávat první verze práce průběžně během roku. Ve dvojicích si budou dávat vzájemnou zpětnou vazbu a dostanou také zpětnou vazbu od učitele. Mohou kapitolu přepracovat podle zpětné vazby. Na konci dotvoří celou práci jako celek.

Zadání maturitní práce

Téma maturitní práce: Tvorba funkční mobilní aplikace pro Android

Termín odevzdání maturitní práce 31. 3. 2020

Délka obhajoby maturitní práce před zkušební maturitní komisí

20 minut (příprava 10 minut)

Požadavek na počet vyhotovení maturitní práce

2 výtisky + pdf a soubor projektu, obě verze je nutné odevzdat nejpozději 31. 3. 2019

Programovací jazyk a prostředí: MIT App Inventor 2

Možné sféry návrhu softwarového projektu:

1. Mapy, navigace, GPS sledování
2. Fotografie, grafika, kreslení
3. Hlasové ovládání, překladač, zvuk
4. Seznamy, časovače, praktické služby
5. Zábava, kvízy, hry

Pokyny k formě maturitní práce

Formát práce musí odpovídat Šabloně písemných prací BMA, která je přílohou tohoto dokumentu.

Styl: Práce je psaná odborným stylem ve spisovném českém jazyce. Text práce je srozumitelný, stručný a výstižný.

Minimální rozsah maturitní práce: 10 normostran (1 normostrana = 1800 znaků včetně mezer)

Maximální rozsah maturitní práce: 25 normostran (1 normostrana = 1800 znaků včetně mezer), do rozsahu se započítává pouze úvod, analýza a závěr.

Struktura maturitní práce

Každý oddíl začíná na nové stránce. Celá práce je vytištěná jednostranně.

1. **Titulní strana** - první list práce, dle univerzální šablony školy
2. **Čestné prohlášení** - na druhém listu práce, text zarovnan vpravo dole, je uvedeno datum odevzdání a vlastnoruční podpis. Přesné znění: "Prohlašuji, že jsem maturitní práci vypracoval/a samostatně s využitím pouze uvedených a řádně citovaných informačních zdrojů."
3. **Obsah** - na třetím listu práce, je strukturován v souladu s členěním vlastního textu práce
4. **Abstrakt** - na čtvrtém listu práce, jde o stručný text shrnující celou práci, rozsah je 150 - 200 slov. Dále je uvedeno 5 až 10 klíčových slov práce. Abstrakt a klíčová slova jsou nejprve uvedena v českém jazyce a následně v anglickém jazyce.
5. **Úvod** - stručný popis, čemu se práce věnuje, zdůvodnění výběru dané filmové postavy k analýze
6. **Analýza** - vlastní text vypracované strukturované analýzy (viz Pokyny k obsahu maturitní práce)
7. **Závěr** - shrnutí nejdůležitějších výsledků analýzy, zhodnocení, jak se podařilo naplnit zadání analýzy, reflexe silných stránek práce a limitů práce, osobní přínos pro autora práce
8. **Informační zdroje** - obsahuje seznam všech citovaných zdrojů podle pravidel citační normy.
9. **Přílohy** - Pokud autor uvádí přílohy, čísluje je ve formě "Příloha č. 1" atd.

Obhajoba

V rámci obhajoby je úkolem žáka předvést vytvořenou aplikaci, shrnout závěry své maturitní práce a popsat nejvýraznější aspekty. Součástí obhajoby je také reflexe celého procesu vývoje softwaru.

Hodnocení maturitní práce

Vedoucí práce a oponent vypracují každý vlastní posudek, ve kterém zhodnotí obsah a formální náležitosti práce. Na základě Pravidel hodnocení a klasifikace žáků ve Školním řádu BMA navrhnou hodnotící známku (s ohledem na váhu jednotlivých kritérií hodnocení). Posudky budou předány žákovi nejpozději 14 dní před termínem jeho obhajoby maturitní práce. O výsledné známce rozhodne maturitní komise s přihlédnutím k oběma posudkům a průběhu obhajoby.

Kritéria hodnocení maturitní práce

20 % Formální náležitosti

Gramatika a stylistika, práce se zdroji, splnění zadání z hlediska formy a struktury práce

60 % Obsah

Splnění zadání z hlediska obsahu vývoje software, kvalita dokumentace, vysvětlení procesu, a odbornost (minimálně) na úrovni vyučované na gymnáziu Beskydy Mountain Academy

20 % Obhajoba

prezentační úroveň, obsahová relevance

V případě závažných nedostatků v relevantnosti formy nebo obsahu k zadání je možné práci hodnotit jako nedostatečnou.

Pozdní odevzdání

Neodevzdá-li žák pro vážné důvody práci ve stanoveném termínu, omluví se písemně řediteli školy nejpozději v den stanovený pro odevzdání maturitní práce; uzná-li ředitel školy omluvu žáka, určí žákovi náhradní termín pro odevzdání maturitní práce.

Pokud žák maturitní práci neodevzdá v termínu bez písemné omluvy s uvedením vážných důvodů nebo pokud mu omluva nebyla uznána, posuzuje se, jako by danou zkoušku vykonal neúspěšně.

Pokyny k obsahu maturitní práce

Obsah musí zahrnovat popis účelu softwaru a problému, k jehož řešení je software určen. Dále musí být zmíněn popis navržených algoritmů a hlavních překážek při implementaci, a také přehled testování a ladění aplikace. Nakonec je nutno zahrnout finální popis celkového procesu. K popisu uživatelského rozhraní a zdrojového kódu by mělo být použito dostatečné množství snímků obrazovky, diagramů a ilustrací.

Struktura analýzy

1. Přehled

- a. Odůvodnění systému
- b. Popis problému, k jehož řešení je systém určen
- c. Očekávaný kontext systému

2. Plánování

- a. Kdo bude systém využívat?
- b. Jak bude systém využíván?
- c. Jaká data by měla být do systému vložena?
- d. Jaká by měla být výstupní data?

3. Návrh

- a. Teoretické schéma systémové architektury, včetně vstupů a výstupů
- b. Popis klíčových algoritmů a datových struktur
- c. Analýza rizik a jejich mitigace

4. Implementace

- a. Podrobný přehled kódovacích metod použitých k implementaci
- b. Diskuze problémů, které se při implementaci objevily, včetně chyb v běžné syntaxi a běhových chyb

5. Testování

- a. Podrobný přehled testovací strategie, včetně testovacích případů a focus groups
- b. Popis ladění aplikace, včetně runtime errors a sémantických chyb
- c. Budoucí rizika a jejich mitigace

6. Reflexe

- a. Procesy a metody užití při vývojovém cyklu
- b. Vývojové prostředí softwaru
- c. Schopnost systému naplnit cíle a vyřešit problémy

Vypracoval: Jonathan Lobel
30.9.2019