# Kvalifikační zkouška – Programátor internetu:

Standardní délka je jeden den. Zkouška probíhá na učebně, vybavené příslušnou technikou, praktická část v místě zkoušky dle pokynů zkoušejícího

Kvalifikační zkouška je určena pro pracovníky, kteří po jejím složení budou připraveni pro činnost na pozici programátora internetu.

Účastníci zkoušky musí znát obecné základy algoritmizace a vytvořit analýzu dané úlohy pro tvorbu programu. Musí umět tvořit a odladit vytvořený program včetně uživatelského rozhraní a otestovat jej. Dále se musí umět orientovat v relačních databázích, znát základní SQL příkazy a jejich využití pro programování. Dále musí umět programovat skripty a dávky v daném programovacím jazyku.

Absolventi této kvalifikace budou mít veškeré kompetence a dovednosti vyplývající z aktuálně platného hodnotícího standardu profesní kvalifikace **18-003-M.**

Pro zájemce o uvedenou problematiku bez základních znalostí a zkušeností, doporučujeme absolvovat příslušné kurzy.

**Zkouška obsahuje tyto témata dle Národní soustavy kvalifikací:**

- **Analýza a algoritmizace praktických úloh**
  - Provést analýzu požadavků a cílů praktického zadání: stanovit jednotlivé kroky vedoucí k řešení daných požadavků a cílů, navrhnout seznam potřebných konstant, proměnných, včetně jejich datových typů – jednoduché (čísla, znaky, logické hodnoty), navrhnout strukturované typy dat (pole, záznam, množina), objekty, jejich rozsah a uložení, stanovit dílčí úkoly (moduly) a navrhnout postup jejich řešení, stanovit parametry, uvést vztahy mezi použitými proměnnými a výpočtové vztahy, uvést použití dílčích úkolů (modulů) v procesu řešení a vztahy mezi nimi, popsat množinu testovacích hodnot pro ověření správnosti algoritmu
  - Vybrat vhodné datové a algoritmické prostředky, sestavit algoritmus a přehledně schematicky vyjádřit (na základě analýzy v předchozím kritériu): popsat strukturu použitých proměnných a konstant, včetně konkrétních použitých datových typů, popsat výpočtové vztahy a další změny dat, popsat strukturu jednotlivých modulů (procedury, funkce, knihovny), popsat použité algoritmické struktury (cykly, podmínky, jednoduché a složené příkazy), sestavit přehledné schéma řešení problému (vývojové diagramy, strukturogramy), stanovit citlivá místa řešení (větvení, cykly) a určit body důležité pro testování správnosti algoritmu
  - Sestavit dokumentaci vytvořeného řešení: vytvořit přehledný zápis jednotlivých požadavků a cílů ze zadání, přiřadit k jednotlivým požadavkům a cílům použité prostředky – proměnné, konstanty, moduly a výpočtové vztahy, stanovit transparentní skupinu testovacích dat s odůvodněním (postihující všechny varianty řešení a testující všechny cesty v navrhovaném algoritmickém řešení)
- **Tvorba programu ve vybraném prostředí**
  - Implementovat vytvořený algoritmus do vybraného programového kódu; vhodně použít datové i programové prostředky vybraného prostředí: přepsat jednotlivé kroky vytvořeného algoritmu (z kompetence „Analýza a algoritmizace praktických úloh“) ve vybraném programovém kódu, definovat proměnné a konstanty pomocí vybraného kódu, definovat moduly pomocí vybraného kódu, použít standardní i vlastní knihovny, doplnit jednotlivé kroky vhodným popisem a poznámkami
  - Odladit vytvořený program: odstranit pomocí kompilátoru syntaktické chyby, odstranit po spuštění významové (sémantické) chyby dosazením vhodných konstant, najít a odstranit případné nevhodné podmínky vedoucí například k nekonečným smyčkám, odstranit chyby podmínek v nastavených cyklech, odstranit nevhodný formát výstupu hodnot (výpis reálných čísel), přehledný výpis textů zlepšující vypovídající hodnotu vystupujících údajů, doplnit komentáře, které dokumentují stav průběhu činnosti programu (např. text „... třídím data“; „... počítám“ apod.)
  - Vybrat vhodný nástroj pro automatizovanou tvorbu dokumentů (JavaDoc, ApiGen atd.) a sestavit programovou dokumentaci: doplnit vytvořený kód programu komentáři k jednotlivým úsekům programu zlepšující čitelnost kódu, zvýšit přehlednost kódu formální úpravou (zarovnání a odsazení) zápisů, zajistit logickou návaznost a zlepšit orientaci v programovém kódu, popsat použité knihovny a jejich umístění

*tel:* 251 551 362
*tel./fax:* 234 697 784
602 308 775, 602 756 640
E-mail: info@jubela.cz

**JUBELA s. r. o.**
**Ing. Marek Laurenčík, MBA**
***Volutová 2523/14 • 158 00 Praha 5***

- Vybrat vhodný nástroj pro automatizované sestavování aplikací (Maven, Jenkins atd.) a vysvětlit principy použití
- Vybrat vhodný nástroj na sledování změn (verzovací systém) ve zdrojových kódech softwaru během jeho vývoje, vysvětlit jeho výhody a nevýhody
- Uložit zdrojovou i binární formu programu a tento postup slovně okomentovat

- **Tvorba uživatelského rozhraní**
  - Vytvořit vhodné uživatelské rozhraní pro komunikaci s programem na základě požadavků stanovených v zadání: vytvořit formulář, případně jiné prostředí pro komunikaci uživatele s programem, umístit do komunikačního prostředí vhodné objekty zvyšující názornost a uživatelský komfort programu, umístit na formulář objekty umožňující výstup dat na obrazovku i tiskárnu, případně objekty umožňující ukončení programu a další prvky pro řízení programu uživatelem (např. formátování dat aj.), vyplnit vytvořený formulář
  - Sestavit dokumentaci pro orientaci ve vytvořeném rozhraní: vytvořit přehledný manuál pro uživatele obsahující popis uživatelského rozhraní, popis funkcí, knihoven (především uživatelských), uvést v dokumentaci technické požadavky programu (především paměťovou náročnost), uvést v dokumentaci kontakt na uživatelskou podporu a kontakt na autory
- **Ověření funkčnosti programu a testování optimálnosti algoritmu**
  - Nastavit testovací data a ověřit funkčnost pro zadanou sestavu vstupních údajů: ověřit jednotlivé části programu použitím testovací množiny dat zvolené v rámci analýzy (z kompetence „Analýza a algoritmizace praktických úloh“), doplnit získané reporty výstupních hodnot jako přílohu k dokumentaci programu. Obhájit navržený postup
  - Testovat optimálnost algoritmu: ověřit časovou náročnost programu použitím vhodné testovací množiny dat zvolené v rámci analýzy, vyhodnotit výsledky testování a opatřit závěrečným komentářem o vhodnosti použití programu, včetně závěrečného zhodnocení splnění zadaných cílů, doplnit k dokumentaci programu
  - Otestovat dílčí funkční testy pomocí jednotkových testů (unit test)
  - Vybrat vhodný způsob šíření k uživateli a zvolit umístění a užití hotového programu na základě požadavků zadavatele: uložit a distribuovat program na datových nosičích (CD, DVD), umístit program na FTP a umístit odkaz na webových portálech, umístit program na webových stránkách zadavatele
- **Orientace v relačních databázích**
  - Vyjmenovat alespoň 5 příkazů jazyka SQL a uvést příklady jejich použití
  - Sestavit netriviální SQL příkaz podle konkrétního zadání
  - Popsat účel zadané části netriviálního SQL kódu a vysvětlit jeho funkci
  - Vysvětlit principy z oblasti popisu datového modelu
- **Základy programování skriptů a dávek**
  - Popsat principy vybraného skriptovacího jazyka (např. VBScript, PowerShell, Bash, Perl, Python); výhody a nevýhody, možnosti využití
  - Naprogramovat jednoduchou úlohu s pomocí zvoleného skriptovacího jazyka (např. hromadné přejmenování souborů, synchronizace adresářů)
  - Využít znalost základních příkazů operačního systému v dávkách a skriptech (např. ls x dir)
  - Porozumět anglicky psanému manuálu, vyhledat v něm požadované informace a využít je k napsání skriptu

*tel:* 251 551 362
*tel./fax:* 234 697 784
602 308 775, 602 756 640
E-mail: info@jubela.cz

JUBELA s. r. o.
**Ing. Marek Laurenčík, MBA**
***Volutová 2523/14 • 158 00 Praha 5***