# POŽADAVKY KE ZKOUŠCE z PRG003

# METODIKA PROGRAMOVÁNÍ a

# FILOSOFIE PROGRAMOVACÍCH JAZYKŮ

Letní semestr 2019

**Zkouška má dvě části**

1. **V první má student prokázat podrobnější znalosti problematiky programovacích jazyků resp. metodiky programování. Může si vybrat, že ji splní jednou z dvou *možností:***

A1) Prokáže při ústní zkoušce podrobnější znalost některého z jazyků probíraných na přednášce. Výběr jazyků je uveden níže. Je možné si individuálně domluvit i jiný jazyk.

A2) Napíše esej na relevantní téma, které si předem nechá schválit přednášejícím
(implicitní délka 10-15 stran, vyberte si problém, který lze v takovém skutečně popsat)
*Pokud budete požadavky zkoušky plnit napsáním eseje,* ***dejte si záležet na formulacích a čitelnosti.*** *Vyplatí se nechat hotový text pár dní „odležet“ a vrátit se k němu tak, že zkusíte zapomenout, že jste autor, a budete ho číst jako nový čtenář.*

Hotovou esej mi pošlete mailem. Pokud bych vám přijetí do dvou či tří dní nepotvrdil, pošlete ho znovu a urgujte odpověď*.* Na přečtení budu zpravidla potřebovat 2-3 dny. Pak se ozvu a dohodneme si konzultaci nad textem eseje.

**Pokud budete psát esej, nepřihlašujte se na vypsané termíny – domluvíme se individuálně.**

1. **Ve druhé při ústní zkoušce prokáže globální přehled o problematice***při napsání eseje k takovým dotazů zpravidla nedojde*

**Ad A1) Podrobnější znalost jednoho z následujících jazyků:**

**( Je možné si domluvit individuálně jiný jazyk – pak se na požadavcích domluvíme)**

**a) SIMULA 67**

pojem třídy - objektové programování

text, předávání parametrů,

kvalifikace a inspekce

prefixování, virtuální atributy, hidden a private, inner,

kvaziparalelní systém - koprogramy

třídy SIMSET a SIMULATION, princip procesní simulace

**b) C++**

realizace myšlenek objektového programování

pojem třídy, členu, konstruktor a destruktor

ochrana přístupu ke členům tříd

vícenásobná dědičnost, abstraktní třídy,

zpracování výjimek, šablony

přetěžování operátorů a metod

srovnání jazyků C a C++

**c) Ada**

charakteristika jazyka

typy a typové kontroly, atributy,

separátní kompilace, práce s knihovnami

generiky

podprogramy, předávání parametrů

výjimky

paralelismus, mechanismus randevous

**d) Java**

koncepce jazyka

applety

realizace myšlenek objektového programování

konstruktory, metoda finalize a garbage collector

abstraktní a finální metody a třídy

rozhraní

vnitřní třídy

ochrana přístupu k členům tříd

balíky

vlákna

příkazy a výrazy

zpracování výjimek

předávání parametrů

automatické dokumentování

**e) C#**

*Jen jako porovnání s Javou resp. s C++*

**f) Python**

*Student, který si vybere Python může dostat úkol naprogramovat jednoduchý problém. Vemte si proto s sebou na zkoušku počítač, kde Python máte.*

Koncepce jazyka

Datové typy, proměnné

Seznamy, n-tice, slovníky

Řetězce

Soubory

Řídící konstrukce

Definice funkcí

Moduly

Objektové programování v Pythonu

Výjimky

**g ) Haskell**

*Nestačí vědomosti, které se učí na přednášce z Neprocedurálního programování.
můžete dostat úkol vytvořit jednoduchý program*

Typy v Haskellu, typová specifikace funkce

Sémantika "mečování" parametrů: proměnné, datové konstruktory, as patterns ( @s ), žolíky ( \_ ), lazy-parametry ( ˜x )

Lazy vyhodnocování, "nekonečné" termy, funkce v Haskellu nejsou striktní.

Lambda abstrakce.

Třídy, podtřídy, instance.

Moduly, abstraktní typy dat

Principy I-O operací, monáda IO

Pojem monády, příklady konkrétních monád.

**Ad B) Obecné otázky**

1. Vývoj programovacích jazyků

FORTRAN, Algol 60, PL/I, Algol 68, Pascal, Simula 67, Ada, C a C++,Java, APL, Python, LISP, Haskell, PROLOG

2. Programovací styly

Strukturované, modulární, objektové programování, event-driven

4. Datové typy a datové  abstrakce, generické typy

5. Řídící struktury programovacích jazyků

6. Struktura programovacího jazyka,

proměnné, jejich hodnoty, typy, definiční oblast, životní cyklus
Překlad a interpretace programovacího jazyka, separátní překlad

7. Podprogramy a předávání parametrů

8. Zpracování výjimek

9. Objektové programování.

Zapouzdření, dědičnost (jednoduchá, vícenásobná a problémy, které přináší), polymorfismus.
Virtuální metody a způsoby jejich implementace, konstruktor, destruktor, „metody tříd“.

Ochrana atributů tříd.

11. Procedurální a neprocedurální programování, logické a funkcionální programování,

lazy vyhodnocování.

**LITERATURA :**

Ghezzi C., Jazayeri,M. : Programming Language Concepts

 John Wiley 1987

Pratt T.W. : Programming Languages - Design and Implementation

 Prentice Halll 1975

 ruský překlad Moskva "Mir" 1978

Horowitz E. (editor): Programming Languages - A Grand Tour

 Computer Science Press 1985 ISBN 0-88175-073-5

SIMULA

Benda, Staudek : Programování v jazyce SIMULA'67

Malík : Programovací jazyk SIMULA skripta MFF UK

ADA

Barnes J.G.P. : Programming in ADA

 Addison-Wesley 1983

C++

Stroustrup Bjarne: C++ Programming Language

 second edition

Addison-Wesley 1993, Reading Massachusets

Lippman Stanley.B. : C++ Primer

second edition

Addison-Wesley 1993, Reading Massachusets

JAVA

Bruce Eckel : Thinking in Java WWW

Pavel Herout : Učebnice jazyka Java, 2001 Kopp

Tomáš Pikner : Java – začínáme programovat, Grada 2002

P.J.Perry : Java – tvorba dokonalých WWW stránek

PYTHON

Daryl Harms, Kenneth McDdonald : Začínáme programovat v jazyce Python Comp. Press 2006 – překlad z angličtiny

Mark Lutz, David Ascher : Naučte se Python, Grada 2003 – překlad z angličtiny