



FAKULTA **ústav**
STROJNÍHO **automatizace**
INŽENÝRSTVÍ **a informatiky**



Různé formy programování robotů a metodika programování

- Připomenout vybrané obecné pojmy z oblasti programování (Opakování je matka učení – lat. přísloví: Repeticio est mater studiówum)
- Zdůraznit význam metodiky programování při návrhu robotických programů
- Upozornit na specifické postupy programování robotů

Softwarové inženýrství definuje programování jako proces návrhu řešení spustitelného programu na zvoleném výpočetním prostředku.

Tento proces zahrnuje činnosti:

- Specifikaci problému
- Analýzu problému
- Nalezení algoritmu k řešení problému (pokud možno optimálního)
- Vytvoření zápisu programu ve zdrojovém jazyku programování
- Pořízení cílového spustitelného kódu programu
- Testování programu zda splňuje dané požadavky
- Sepsání dokumentace k programu
- Následnou údržbu při používání programu

Protože sestavení a následný zápis potřebného množství strojových instrukcí v binárním tvaru je velmi pracné a dochází při něm k mnoha chybám, využívá se myšlenka automatizovaného získání strojového kódu programu pomocí kompilátoru.



Algoritmus je přesný návod či postup, kterým lze vyřešit daný typ úlohy.
Pojmem **algoritmus** se při programování označuje *teoretický princip řešení* problému (oproti přesnému zápisu v konkrétním programovacím jazyce). (SE wiki)

Pozn. Popis algoritmu lze provést různými způsoby (Použil někdo např. vývojové diagramy viz norma ČSN ISO 5807?)

Pod pojmem **programovací jazyk** rozumíme prostředek pro zápis algoritmů, jež mohou být provedeny na počítači. Je komunikačním nástrojem mezi programátorem, který v programovacím jazyce formuluje postup řešení daného problému, a počítačem, který program interpretuje technickými prostředky. (SE wiki)

Výrazným rysem současných programovacích jazyků je možnost, opakované problémy definovat jako podprogramy, rozdělit program na logické bloky, a postupovat v jeho sestavování metodou TOP DOWN (shora dolů), případě nutnosti i zdola nahoru (BOTTOM UP).

(Preferujte postup shora dolů!!!)

Protože úlohy, řešené pomocí počítačů jsou různé, programovací jazyky jsou většinou **problémově orientované** na určitý druh algoritmů (např. matematicko-technické výpočty, hromadné zpracování dat, programování úloh umělé inteligence, ovládání numericky řízených obráběcích strojů, řízení soustav v reálném čase, zpracování textových seznamů, atd. a konec konců i robotů.

Při řešení úloh je potřeba nejen definovat
postup řešení, ale také **datové údaje**,
které se mají zpracovat!

Proto programovací jazyky musí mít
prostředky pro definování a používání
různých **datových struktur!!!** A to jak
konstant, tak **proměnných**!

Zápis zdrojového „textu“ programovacího jazyka může být řešen formou:

- Alfnumerického textu (nejčastější)
- Manipulací s ikonami (vizualizační programování – dnes využívané)
- Kreslením obrázků a skic

Pracuje se na:

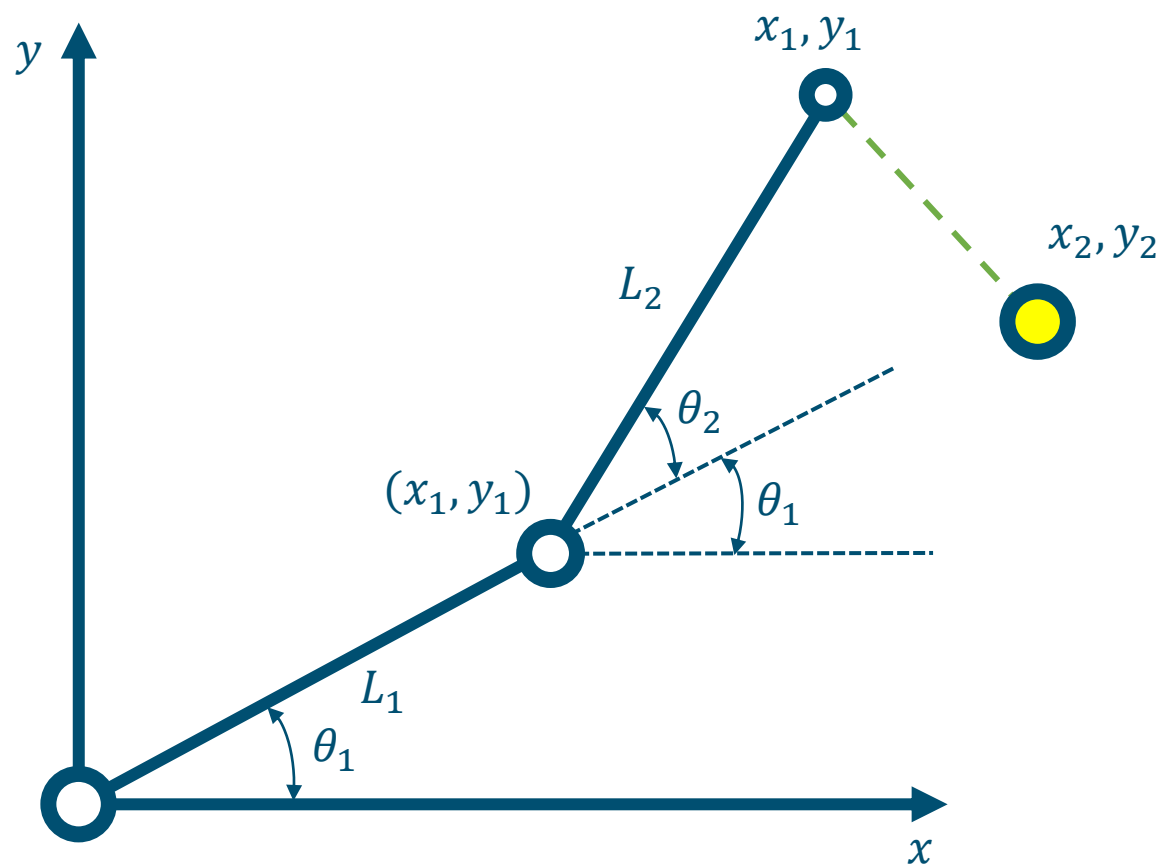
- Kopírování dodaného vzoru (naučením dodaného vzoru)
- Porozumění slovních příkazů
- Porozumění gestům

V praxi dochází často ke vzájemnému kombinování těchto forem!!!

Je zajímavé, že způsob programování prostřednictvím „**kopírování dodaného vzoru (naučením dodaného vzoru)**“, který se v minulých programovacích jazycích nevyskytoval, patří k zásadním způsobům programování robotických soustav (např. průmyslových robotů pro nanášení barev stříkacími pistolemi), ale zejména současných kolaborativních robotů.

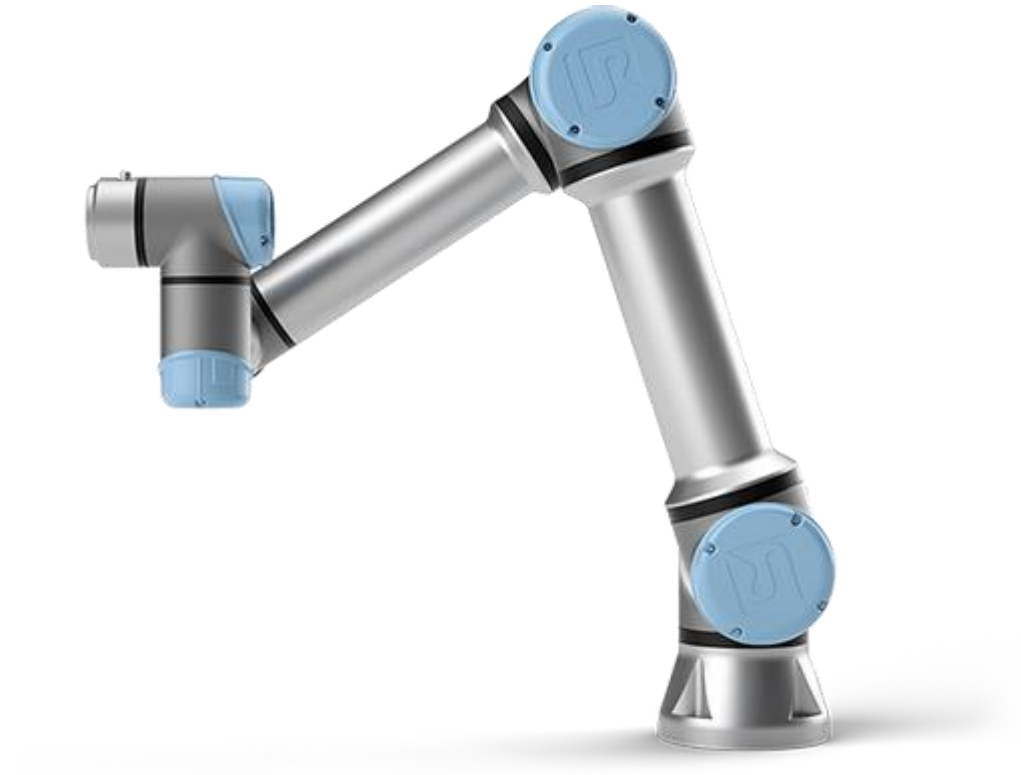


Move (X1,Y1) TO (X2,Y2)



Nebo s koncem ramene ze stávající
polohy pohnete do cílové polohy ramene!
Musíte ale zajistit, aby si zapamatoval tu
výchozí a tu cílovou!
No a pak zmáčknete tlačítko:
Proved' co se „naučil“.

Co je to kolaborativní robot?



Imperativní programování

klasický způsob řazení jednotlivých příkazů, které se mají postupně vykonat, aby se problém vyřešil

Deklarativní programování

popisuje se, čeho se má dosáhnout -ne jak. Pokud se používá jen výčet funkcí, označuje se to jako *funkcionální programování* (LISP – lambda kalkul), pokud vyjadřujeme ní fakta a logické vztahy mezi nimi, označuje se to jako *logické programování* (PROLOG – predikátová logika prvního řádu). Pokud se deklaruje jen seznam problémů, které se mají řešit mluvíme o *problémově orientovaném programování*.

Objektově orientované programování (dnes nejpoužívanější)

pracuje s definovanými objekty a využívá jejich různých vlastností (SE wiki)

Program je potřeba nejen vytvořit, ale také **otestovat**, zda kvalitně splňuje řešení požadovaného problému a dalších na něj kladených požadavků!!! (Ladění programu:-)

Testováním lze prokázat, že chyby, které program má.

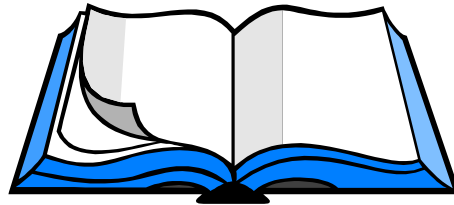
Testováním nelze dokázat, že program chyby neobsahuje!!!

(Na řešení tohoto problému se pracuje! :-)

Softwarové inženýrství vytvořilo řadu obecných doporučení a řadu propracovaných metod, jak správně postupovat při tvorbě programů.

Mluvíme o tzv. obecnějších metodikách nebo konkrétnějších metodách tvorby programů

Popsaný, ověřený a doporučovaný postup optimálního způsobu řešení určitého problému.



Používání metod v je považováno za znak profesionality!!!

Pozor na odlišný přístup k metodám v ČR!



Nejznámější jsou:

Ole-Johan Dahl, Edsger Wybe Dijkstra, Tony Hoare: *Structured Programming*. Academic Press, London 1972 – pro návrh programů klasických procedurálních jazyků: např. Python a C

Návrh objektově orientovaných programů založený na modelovacím jazyku UML pro C++ (Arlow – Neustad: Unifikovaný proces vývoje aplikací, Compter Press 2007)

V současné době jsou to metody agilního přístupu k programování např. SCRUM (pro C++, Python, Java)

DOPORUČENÍ: Používejte pokud možno některou metodu! Vyhnete se mnoha chybám, které vám budou následně způsobovat těžkosti

Zpřesnění:

Pro účely této přednášky se budeme zabývat programováním průmyslových robotů! Tedy pomineme problematiku programování humanoidních robotů, androidů, mobilních robotů (pozemních i dronů) a speciálních robotických prostředků (SW roboty).

První průmyslové robotické systémy byly konstruovány jako jednoúčelová zařízení (tvrdá automatizace) pro přesně vymezené úkony s určitým výrobkem! Většinou se jednalo o manipulátory. Právě nutnost, širšího využití pro jiné výrobky, vedla k požadavkům na pružnou změnu funkcí s rozličnými součástkami, vedla k myšlence vzniku programovatelných robotických soustav

S ohledem na tehdy používané elektrické, pneumatické, hydraulické pohony, a s ohledem na analogové prostředky automatické regulace, se ke změně funkcí a nastavení činnosti těchto prostředku používalo tvarovaných vyměnitelných vaček, různě posouváných zářezek a koncových spínačů, různých přepínačů, nastavitelných potenciometrů apod. Tyto prostředky umožňovaly nastavit takový stroj pro různé výrobky a funkce.

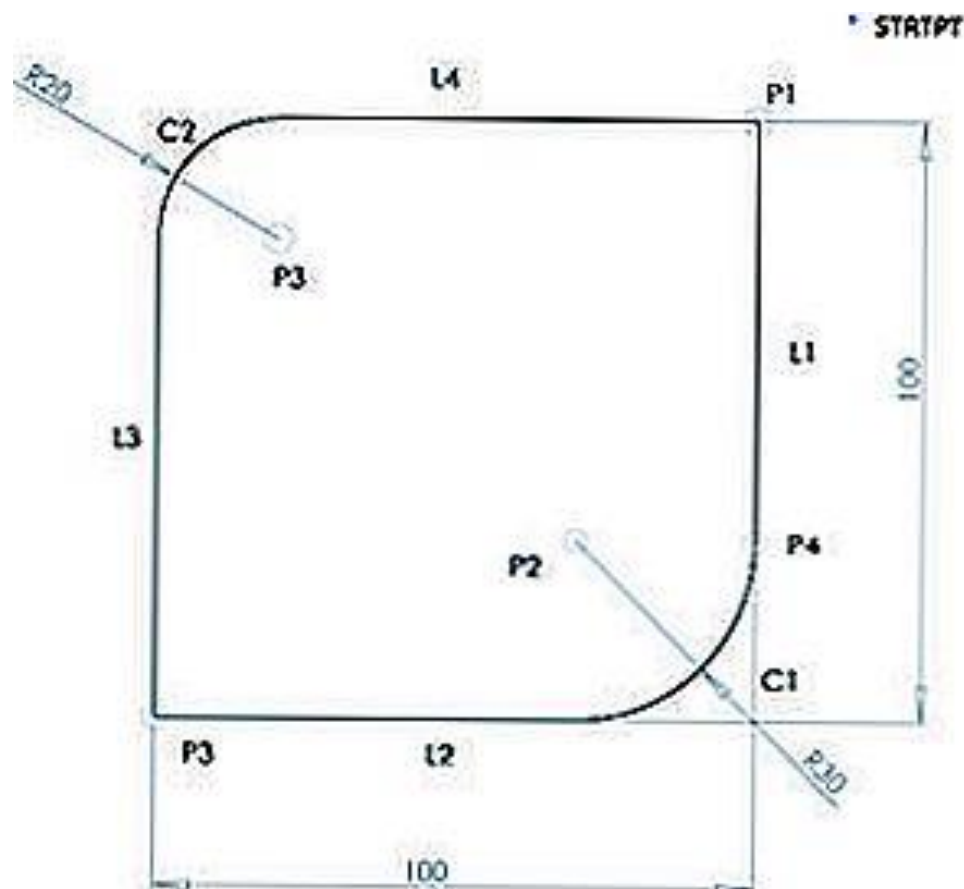
Teprve pokroky v konstrukci elektrických pohonů a digitalizace řídicích soustav s využitím číslicové mikroprocesorové techniky v osmdesátých letech umožnily přejít na řízení pomocí počítačových programů i u robotů.

Přitom vývoj robotických soustav probíhal paralelně s vývojem numericky řízených obráběcích strojů a mikropočítačů, který byl doprovázen vývojem nových čidel a výkonných akčních prvků, počítačové sítě (drátových i bezdrátových).

Technické pokroky v těchto zmíněných oblastech umožňovaly i měnit způsoby programování, zejména ve smyslu jeho zjednodušování, protože řadu řídicích funkcí na sebe „dokázaly převzít“ technické prostředky využívající různé integrované digitalizované prostředky. (Inteligentní čidla a senzory, inteligentní servomotory apod.)

První programovací jazyky pro průmyslové roboty se proto podobaly programovacím jazykům pro numericky řízené obráběcí stroje jako např. byl zprvu alfanumerický zápis jednotlivých instrukcí NC strojů a později např. jazyk APT (Automatic Programming of Tools) a jeho varianty ADAPT, EXAPT a další.





SAAB-ADAPT DATUM 28. 1.1971

```
1  PARTNO | CLPRNT, EXAM1 TOS=K
2  CLPRNT
3  )INTPOL/LINEAR, /
4  CUTTER/D
5  L1=LINE/(POINT/10,0), ATANGL, 90
6  L2=LINE/PARLEL, L1, XLARGE, 70
7  C1=CIRCLE/40,10,50
8  FEDRAT/100
9  FROM/0,0,0
10 GO/TO, L1
11 GOLFT/L1, PAST, C1
12 GOFWD/C1, ON, L2
13 AUXFUN/02
14 END. FINI
```

Jazyk APT, byl mezinárodně standardizován ale řada robotických firem (ABB, KUKA, FANUC a další), financovaly vývoj vlastních podobných jazyků, které se dodnes v různých firmách k jejich robotům stále používají.

Od takto imperativních příkazů včetně definované podoby dráhy nástroje u obr. strojů – u robotu dráhy konce ramene, která byla pracně zadávána v souřadnicích, se přešlo k možnosti odvodit dráhu ramene převzetím/odvozením této dráhy z programu, který byl vytvořen **prostředky CAD/CAM nebo získán digitálním skenerem.**

Ještě obecnou poznámku:

Jestli dokážete mikropočítač propojit s nějakým robotickým zařízením, pak je pouze otázka, jak pořídit kompilátor vámi vybraného jazyka pro použití mikropočítač, aby generoval ovládací příkazy pro připojený robotický systém. Tam kde se vám nedostává běžných prostředků použitého programovacího jazyka, použijete jednoduchý trik – **vypracujete si speciální knihovnu parametrických podprogramů a funkcí!** Často takové knihovny můžete stáhnout z různých portálů robotických firem nebo stránek, podporující konkrétní programovací jazyk. Toto se v řadě případů používá!

OFFLINE ROBOT PROGRAMMING

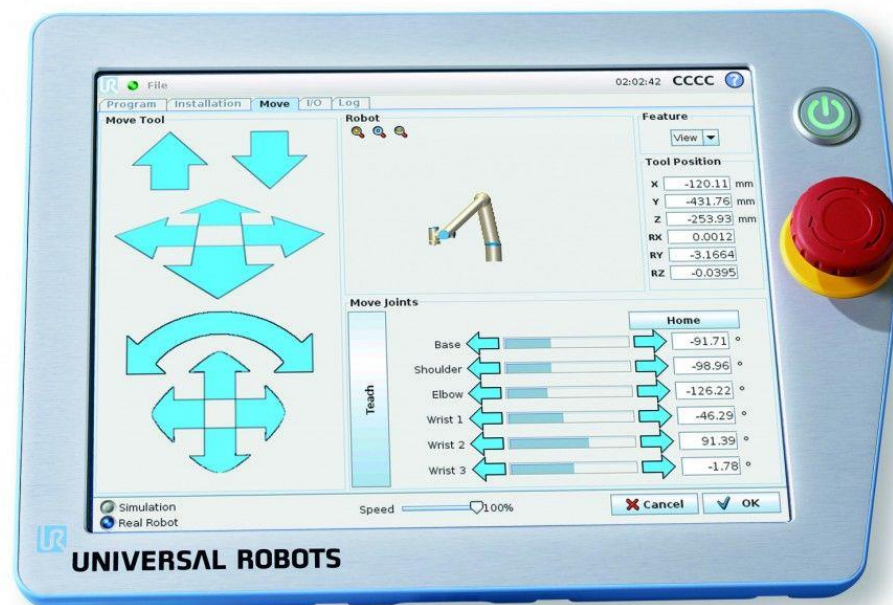
programování se děje mimo robotickou soustavu na nějakém vhodném počítači, kde je nainstalován příslušný software na psaní zdrojového programu, kompilátor, editor pro opravu chyb zdrojového programu, potřebné knihovna podprogramů, atd. A pak se program vloží do řídicího systému robota, testuje se a posléze používá ve výrobě.

ONLINE ROBOT PROGRAMMING

programování se děje **přímo na spuštěném robotu** a využívá se principu učení (teachin) prostřednictvím předvedených potřebných pohybů a činností, s využitím podpory tzv. **teach-pendantů**

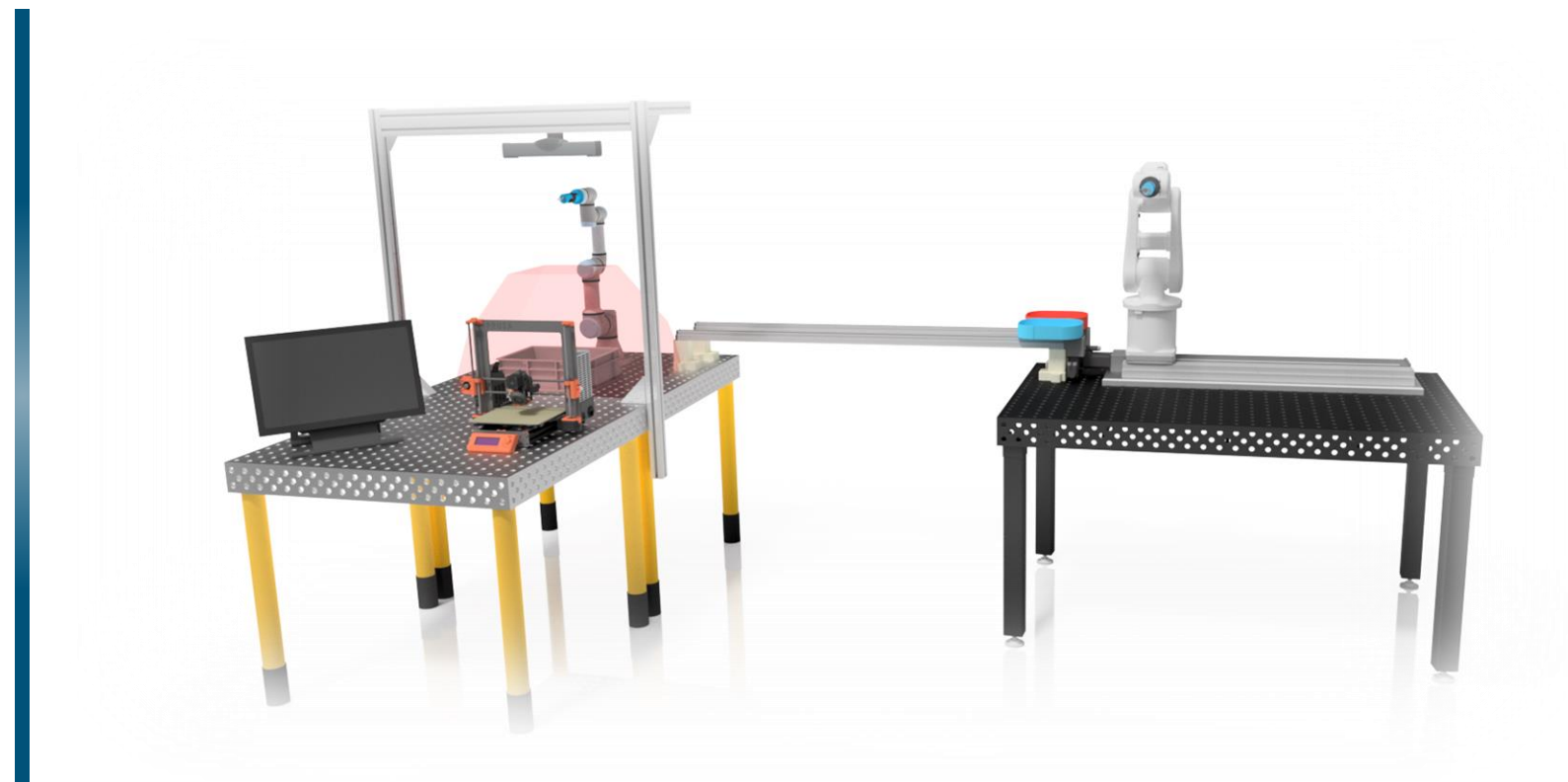
Tento poslední princip programování robotů bylo využívá principu učení, kdy se robotické soustavě „ukáže, co má robot s ramenem udělat“ přímým ručním pohybováním ramene do příslušných poloh a „ukázka“ se „doplní přes teach-pondant“, což je specializovaný on-line propojený tablet (často dotykový), a to ve zvláštním režimu „učení“, který zajistí, aby si to vše robot zapamatoval a pak následně provedl.

Toto „on-line programování“ si ukážete a vyzkoušíte ve cvičeních tohoto předmětu.

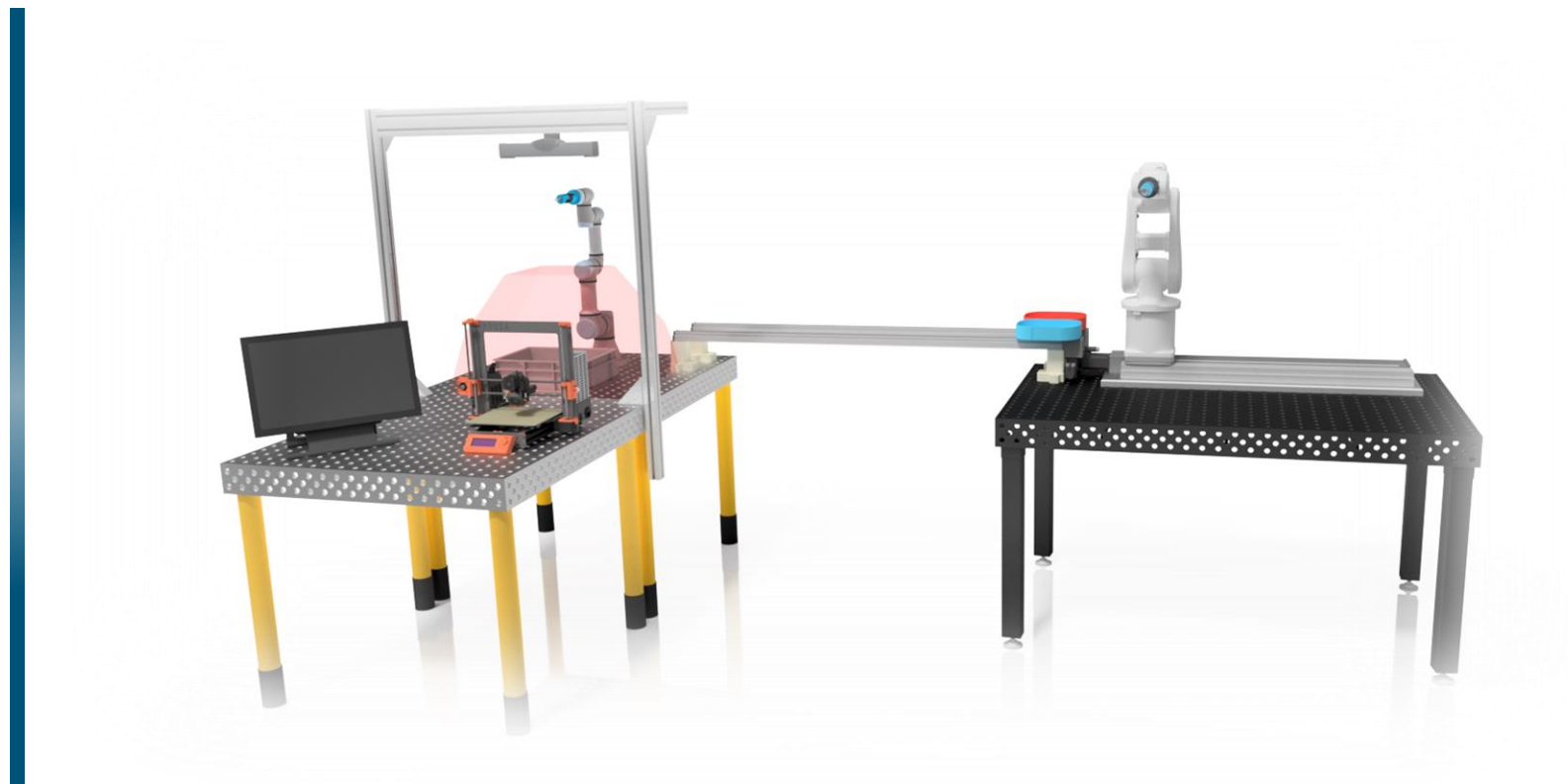


**Přeji hodně rozvahy, intuice,
pozornosti , nadšení a
schopnosti systémového
myšlení při programování
robotů nejen v naší laboratoři,
ale i v pozdější praxi.**

Děkuji za
pozornost!



Otázky?





FAKULTA **ústav**
STROJNÍHO **automatizace**
INŽENÝRSTVÍ **a informatiky**