

# ZÁKLADY PROGRAMOVÁNÍ

Mgr. Vladislav BEDNÁŘ

- 2013
- 2+2.1
- 3/14



Co je vhodné vědět, než si vybereme programovací jazyk a začneme programovat roboty.

# ROBOTI

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZÁBAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

## Programování na PC

- Při programování na PC musíme nejdříve řešit, pod jakým operačním systémem budeme programovat a na jakém systému dotyčná aplikace bude provozována.
- Příkladem může být programování v Javě, kdy dotyčnou naprogramovanou aplikaci můžeme přenést třeba z klasického počítače s operačním systémem Windows na mobilní telefon pracující s operačním systémem Android.



# ROBOT

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



- Pokud známe dokonale syntax programovacího jazyku, můžeme programovat přímo v textovém rozhraní např. editorem PSP.
- Toto je ale vhodné pouze pro tvorbu menšího projektu či zdrojového kódu. U větších projektů je vhodnější použít specializované programátorské rozhraní.

```
1 IDENTIFY HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2 <html>
3 <head>
4 <meta http-equiv="content-type" content="text/html; charset=windows-1250">
5 <meta name="generator" content="PSPad editor, www.pspad.com">
6 <title></title>
7 </head>
8 <body>
9
10 </body>
11 </html>
12
```

```
1 ( replace all occurrence of "MyObject" with your object name )
2 unit (Unit_name);
3
4 interface
5
6 uses
7   Windows, Messages, SysUtils, Classes;
8
9 type
10   TMyObject = class(TObject)
11   private
12     { Private declarations }
13   public
14     constructor Create;
15     destructor Destroy; override;
16     { Public declarations }
17   end;
18
19 var
20   MyObject: TMyObject;
21
22 implementation
23
24 { Object constructor }
25 constructor TMyObject.Create;
26 begin
27   inherited Create;
28   { enter your code here }
29 end;
30
31 destructor TMyObject.Destroy;
32 begin
33   { enter your code here }
34   inherited;
35 end;
36
37 end.
38
```

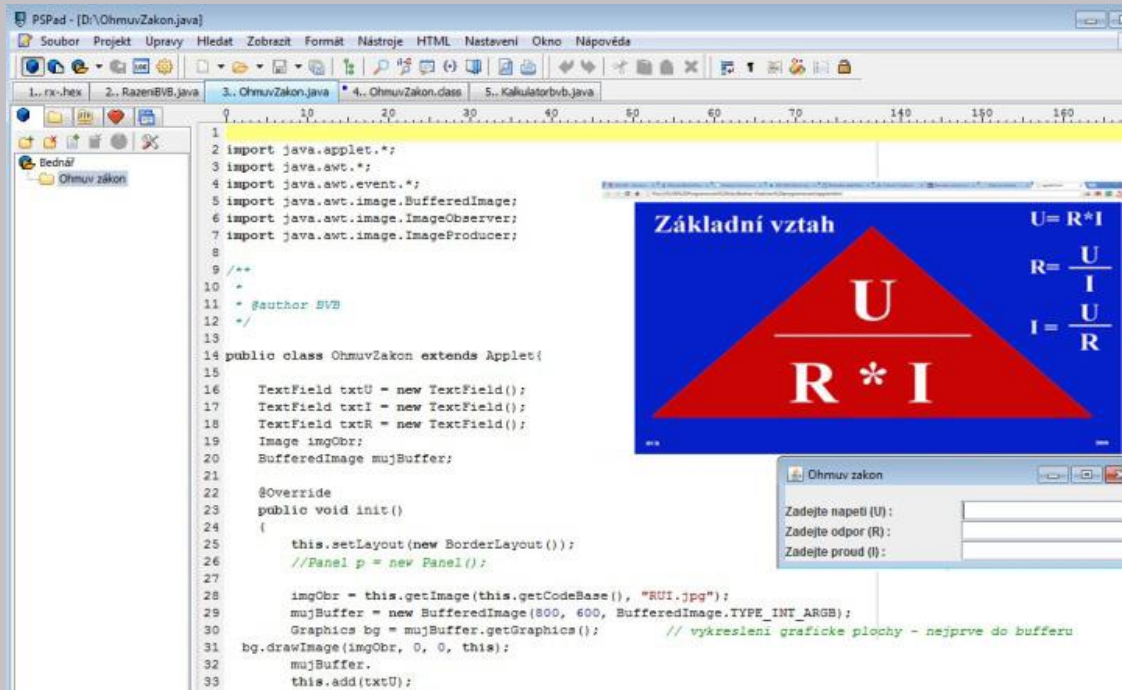
# ROBOTI

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



## Ukázka JAVA appletu

- Java applet je program spustitelný přes webový prohlížeč.
- V počítači musíme mít nainstalované javové rozhraní.



# ROBOT

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU

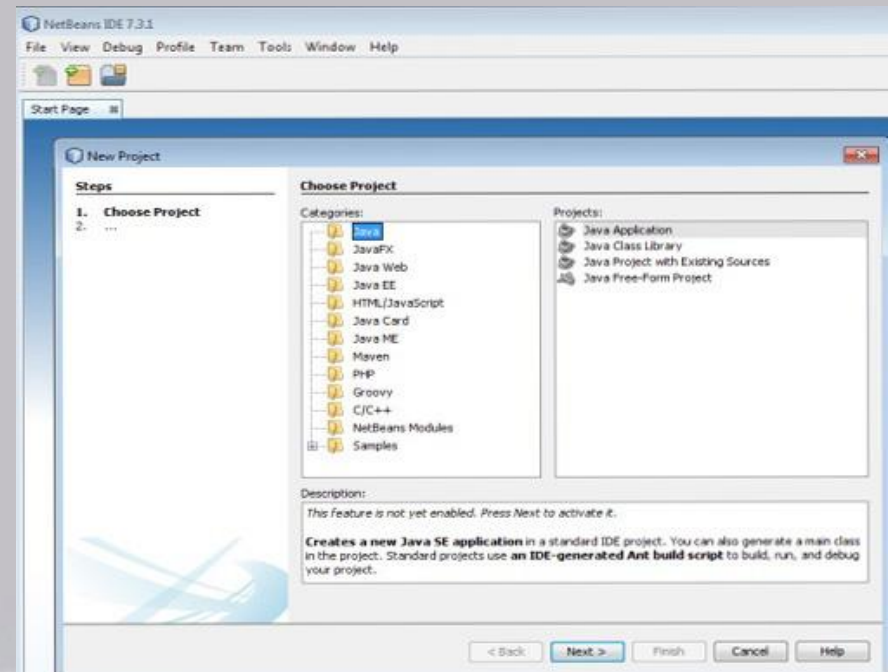


## Vývojová programátorská prostředí

- Pro psaní programu je vždy výhodnější pracovat v nějakém specializovanějším vývojovém prostředí. Příkladem může být Eclipse, NetBeans.

- Vývojová prostředí (IDE), jsou určena na psaní, překlad a ladění aplikací.

Tato vývojová prostředí jsou primárně v jazyce Java, ale podporují skoro jakýkoliv programovací jazyk.



# ROBOTI

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZÁBAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

- V dnešní době získávají na popularitě služby vývojových prostředí, IDE v cloudu. Cloud jsou vlastně internetové služby programů, jež nejsou přímo v našem počítači, ale někde na vzdáleném serveru, a my využíváme jeho služeb jak pro programování, tak i výsledky naší činnosti (vytvořený program) se mohou uchovávat mimo nás.
- Do cloudu se pomalu stěhuje také vývoj softwaru, především mobilních aplikací. Cloudové služby lákají vývojáře do cloudu a slibují jednoduché použití, možnosti spolupráce, neomezený přístup odkudkoliv, a to i z několika počítačů.
- Pro práci v cloudu musí být programátor jen připojen k internetu a vytváří aplikace na vzdáleném serveru. Nemusí mít nainstalované ve svém počítači žádné vývojové prostředí.



# ROBOTI

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



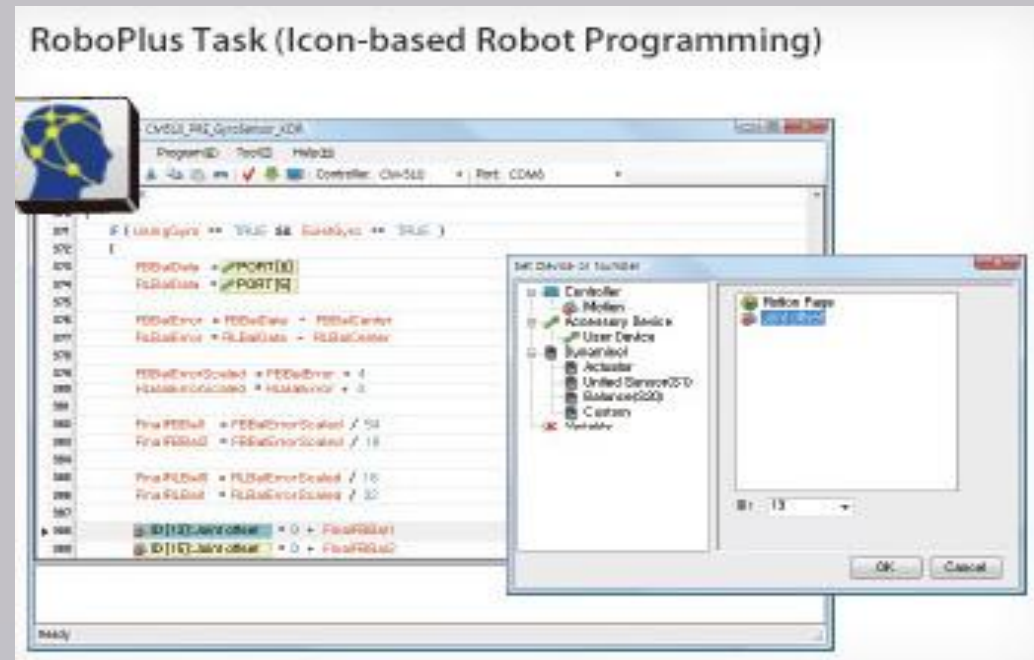
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

## Programování robotů Robotis BIOLOID

- Pro práci s výukovou stavebnicí a programování RoboPlus dodává výrobce software BIOLOID Premium kit.

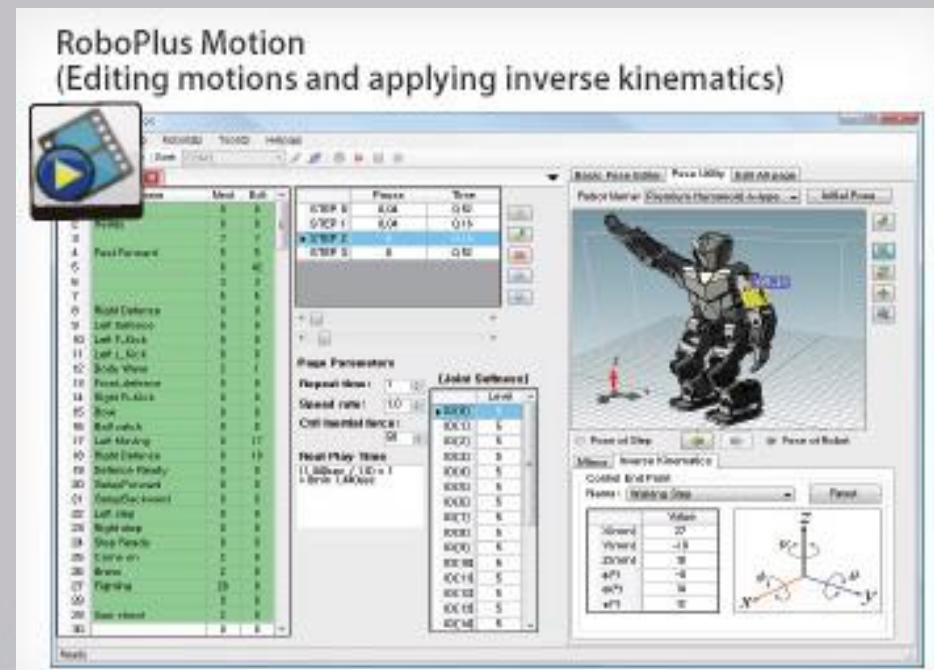
- RoboPlus Task**

Jde o softwarový nástroj, který umožňuje přímou komunikaci s robotem v textovém režimu. Uživatelé zde mohou vytvářet své vlastní příkazy a definovat instrukce.



## RoboPlus Motion - inverzní kinematika

- Pomocí tohoto programu můžeme nepřímo vytvářet program pomocí mechanických pohybů robota, které zaznamenáváme programem
- RoboPlus Motion**  
Program si ukládá výchozí pozice, které potom může vykonávat. Jde o editor pohybu, který můžeme nazývat inverzní kinematikou. Při tomto programování nemusíme znát žádný programovací jazyk.





# ROBOTI

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

## RoboPlus Manager

- Je určen pro správu všech funkcí robota. Tento software sleduje a nastavuje správu firmware.
- RoboPlus Manager poskytuje jednotný pohled na všechny součásti (komponenty a snímače) připojené na komunikační sběrnici. Můžeme zde upravovat parametry každého prvku v reálném čase.

RoboPlus Manager  
(Managing controller and Dynamixel firmware)

Address	Description	Value
24	Torque Enable	0
25	LED	0
26	CW Compliance Margin	1
27	CCW Compliance Margin	1
28	CW Compliance Slope	50
29	CCW Compliance Slope	50
30	Goal Position	265
32	Moving Speed	0
34	Torque Limit	1023
36	Present Position	265
38	Present Speed	0
40	Present Load	0
42	Present Voltage	12.7
43	Present Temperature	35
46	Moving	0

Connection Complete (Controller Firmware Version : 1.1962)



# ROBOTI

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



## Examples in C Language

- Umožňuje vývoj autonomního chování robotů. Příkazy se zadávají velmi podobně jako v programovacím jazyku C..
- Je určen jak pro programátory v jazyku C, tak pro uživatele, kteří se teprve setkávají poprvé s objektovým programováním.

### Examples in C Language (Learning robot control using C language)

```
int del_fakt_sad(unsigned char *pbuffer, int number) {
    // transmitting data
    // comment: data is being sent
    // comment: number of data transmitted is to given
    int count = number;
    while(count > 0) {
        *pbuffer++ = number;
        count--;
    }
}

int del_fakt_sad(unsigned char *pbuffer, int number) {
    // receiving data
    // comment: data is being received
    // comment: number of data received is to given
    int count = number;
    while(count > 0) {
        *pbuffer++ = number;
        count--;
    }
}
```

# ROBOTI

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

## INTERNETOVÉ ZDROJE DOPORUČENÉ K NAHLÉDNUTÍ

<http://www.pspad.com/cz/>

[https://netbeans.org/index\\_cs.html](https://netbeans.org/index_cs.html)

[http://vyuka.pecinovsky.cz/vse/115/IDE\\_NetBeans\\_S.htm](http://vyuka.pecinovsky.cz/vse/115/IDE_NetBeans_S.htm)

[http://www.panrepa.org/CASE/jaro2007/ide\\_case\\_jaro2007.pdf](http://www.panrepa.org/CASE/jaro2007/ide_case_jaro2007.pdf)

## Použité zdroje

ŠVÍGLER, Jan. Algoritmizace a řešení problémů: Vliv typu úlohy na programování. 6.12.2008.

Bioid: Projekty robotů do škol. Robotis: RoboPlus and C Language Solution [online]. [cit. 2013-05-11]. Dostupné z: [http://www.robotis.com/x/BIOLOID\\_main\\_en](http://www.robotis.com/x/BIOLOID_main_en)



# ROBOTI

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

# Konec

## Děkuji všem přítomným za pozornost.

**Autor** : Vladislav Bednář  
**Kontakt** : [bednar@sse-najizdarne.cz](mailto:bednar@sse-najizdarne.cz)  
**Vytvořeno** : 14. 12. 2013

**Střední škola elektrotechnická, Ostrava,  
Na Jízdárně 30, příspěvková organizace**

