

# Přednáška 6

## Algoritmické myšlení II

CORE013 Vývoj softwarových systémů: od myšlenky k funkčnímu řešení

## 6. Algoritmické myšlení II

- [MIT App Inventor](#)
- Interaktivní demo na vytvoření chatovací aplikace

### Domácí práce a příprava na tuto přednášku

- Prosím přineste si vlastní notebooky a na své chytré telefony (pokud máte) si nainstalujte MIT AI2 Companion
- <https://appinventor.mit.edu/explore/ai2/setup-device-wifi>

# VÝVOJ APLIKACE PRAKTICKY

# Vývoj chatovací aplikace

- Máme za úkol vyvinout základní chatovací aplikaci pomocí MIT App Inventoru.
- Po nainstalování této aplikace uživatel zadá své jméno a poté pod ním může posílat textové zprávy.
- Tyto zprávy se zobrazují jako seznam na obrazovce a zároveň se ukládají do databáze.
- Při každé aktualizaci databáze se aktualizuje i seznam zpráv na obrazovce aplikace (tím se uživateli zobrazí nově příchozí zprávy).

# Vývojové prostředí - <http://ai2.appinventor.mit.edu/>

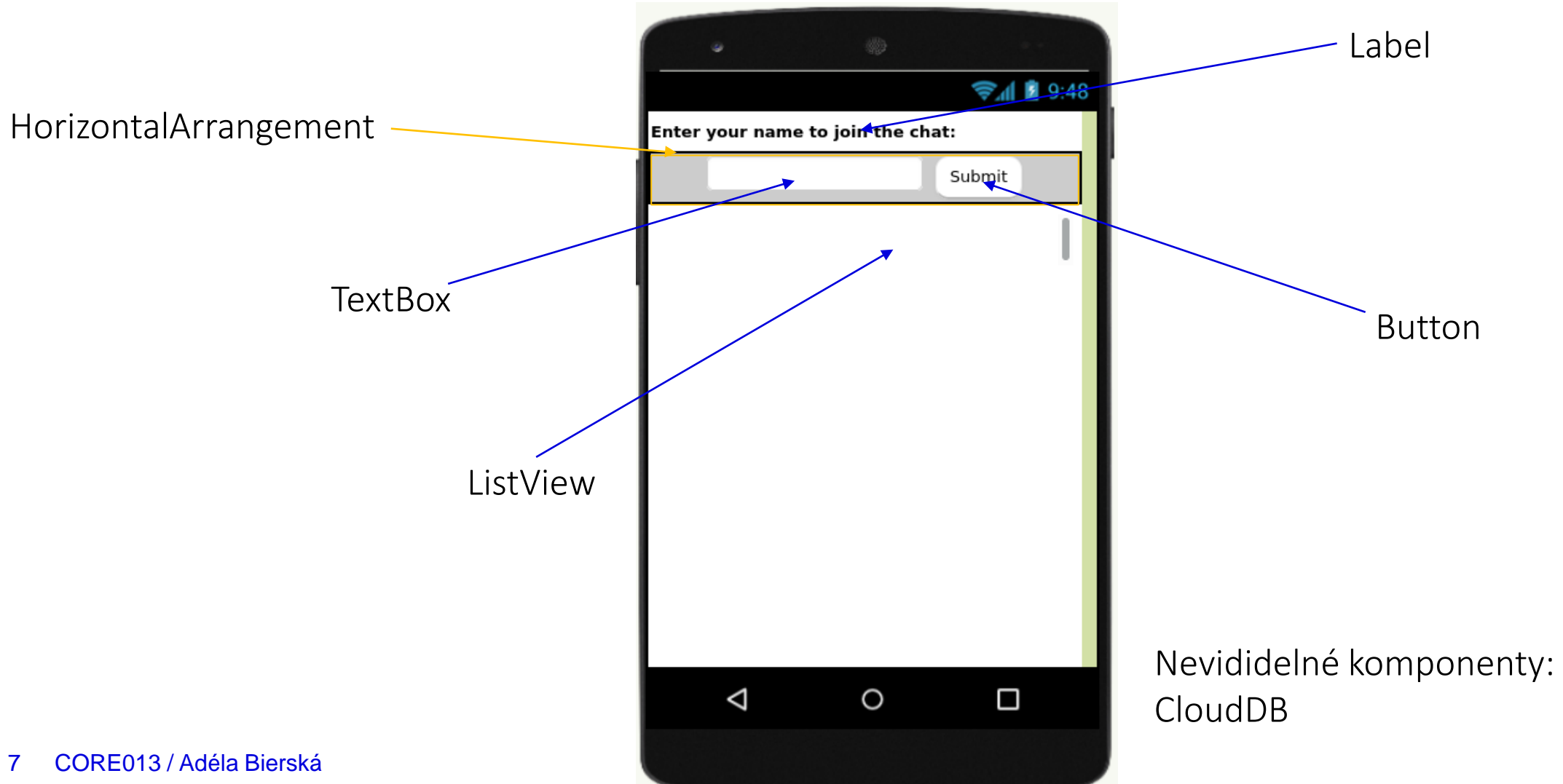
The image shows the MIT App Inventor web-based development environment. The interface is divided into several sections:

- Top Bar:** Contains the MIT App Inventor logo, navigation menus (Projects, Connect, Build, Settings, Help), and user information (My Projects, View Trash, Guide, Report an Issue, English, 493121@mail.muni.cz).
- Chat:** A green bar at the top left with a "chat" label and buttons for "Screen1", "Add Screen...", "Remove Screen", and "Publish to Gallery".
- Designer:** The main workspace, split into three panels:
  - Palette:** A list of UI components categorized into "User Interface" (Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, Switch, TextBox, TimePicker, WebViewer), "Layout", "Media", and "Drawing and Animation".
  - Viewer:** A central area showing a mobile phone simulation. It includes a checkbox for "Display hidden components in Viewer" and a dropdown for "Phone size (505,320)". The phone screen displays "Screen1".
  - Components:** A panel on the right showing the current screen ("Screen1") and buttons for "Rename" and "Delete".
  - Properties:** A panel on the far right showing the properties for the selected component. Properties include: AboutScreen (text input), AccentColor (Default), AlignHorizontal (Left: 1), AlignVertical (Top: 1), AppName (chat), BackgroundColor (Default), BackgroundImage (None...), BigDefaultText (checkbox), BlocksToolkit (All), CloseScreenAnimation (Default), DefaultFileScope (App), HighContrast (checkbox), and Icon.

# Vývojové prostředí - <http://ai2.appinventor.mit.edu/>

The image shows the MIT App Inventor web interface. At the top, there is a navigation bar with the MIT App Inventor logo on the left and menu items: Projects, Connect, Build, Settings, and Help. On the right side of the top bar, there are links for My Projects, View Trash, Guide, Report an Issue, English, and a user email address (493121@mail.muni.cz). Below the top bar is a green header area with a chat icon on the left and buttons for Screen1, Add Screen..., Remove Screen, and Publish to Gallery. On the far right of this header are buttons for Designer and Blocks. The main workspace is divided into three sections: 1. Blocks: A vertical sidebar on the left containing categories of code blocks: Built-in (Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, Procedures), Screen1 (Label1), and Any component. At the bottom of this sidebar are Rename and Delete buttons. 2. Viewer: A central area showing a list of available blocks for the current screen, including 'if then', 'if then else', 'if then else if then else', 'for each number from 1 to 5 by 1', 'for each item in list', and 'for each key with value in dictionary'. 3. Canvas: A large white area on the right where code blocks are assembled. The current code block is a 'when Screen1.Initialize' block containing a 'do' block with two sub-blocks: 'initialize local variable to "Hola Mundo!"' and 'set Label1.Text to get variable'. On the right side of the canvas, there is a teal backpack icon and a vertical toolbar with icons for zooming in (+), zooming out (-), and a trash can icon.

# Rozvržení a komponenty



# Rozvržení a komponenty

Kategorie	Komponenta	Název	Vlastnost	Nastavení
User Interface	Label	Label1	Text	“Enter your name to join the chat”
Layout	HorizontalArrangement	HorizontalArrangement1	Width AlignHorizontal	“Fill Parent” “Center: 3”
User Interface	TextBox	TextBox1		
User Interface	Button	SubmitButton	Text	“Submit”
User Interface	ListView	ListView1	BackgroundColor TextColor	“White” “Black”
Storage	CloudDB	CloudDB1		



# Po spuštění

```
initialize global userName to ""  
initialize global joined to false  
initialize global chatList to create empty list
```

# Přihlášení uživatele

```
when SubmitButton .Click
do
  if not get global joined
  then
    if not isEmpty TextBox1 . Text
    then
      set global userName to TextBox1 . Text
      set global joined to true
      set Label1 . Text to "Type in your message and press Submit to send..."
```

# Odeslání zprávy



# Submit Button

```
when SubmitButton .Click
do
  if not get global joined
  then
    if not isEmpty TextBox1 . Text
    then
      set global userName to TextBox1 . Text
      set global joined to true
      set Label1 . Text to " Type in your message and press Submit to send... "
    else
      call CloudDB1 .AppendValueToList
      tag " message "
      itemToAdd join get global userName
      " : "
      TextBox1 . Text
    set TextBox1 . Text to " "
```

# Zobrazení zpráv

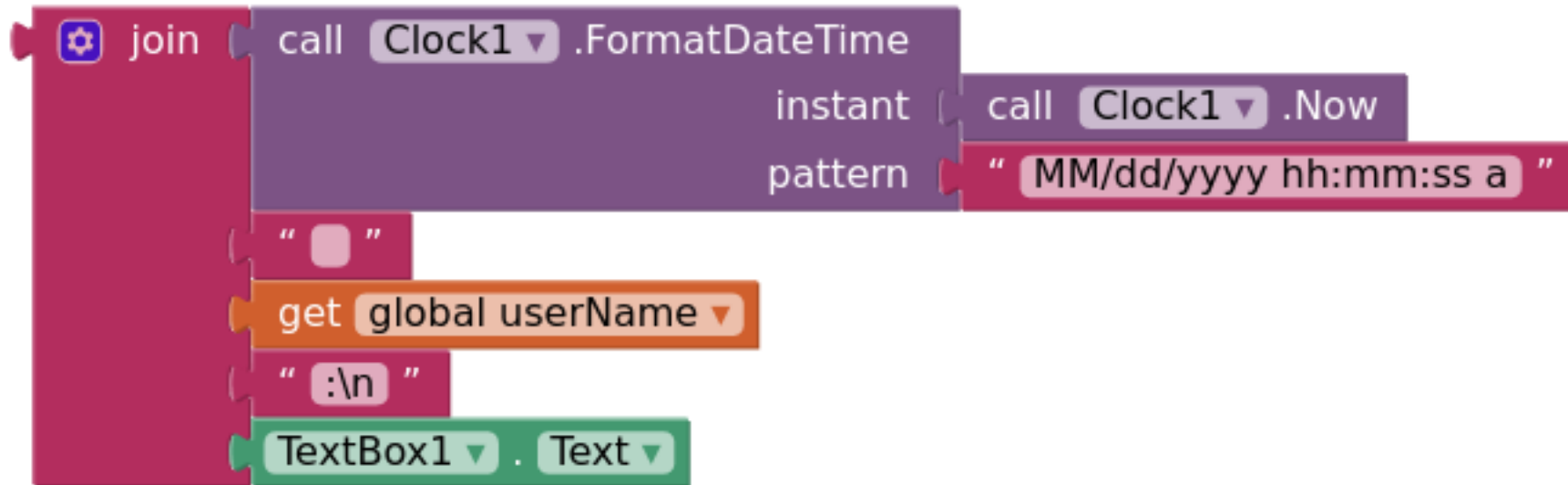
```
when Screen1 ▾ .Initialize
do
  call CloudDB1 ▾ .GetValue
  tag "message "
  valueIfTagNotThere create empty list
```

```
when CloudDB1 ▾ .GotValue
  tag value
do
  if
  get tag ▾ = "message "
  then
  set global chatList ▾ to get value ▾
  set ListView1 ▾ . Elements ▾ to get global chatList ▾
```

# ROZŠÍŘENÍ

# Časové značky

- Je třeba v Designer módu přidat Sensors -> Clock



# Lokální uložení jména

- Je třeba v Designer módu přidat Storage -> TinyDB

```
when SubmitButton .Click
do
  if not (get global joined)
  then
    if not (is empty (TextBox1 . Text))
    then
      set global userName to (TextBox1 . Text)
      set global joined to true
      call TinyDB1 .StoreValue
        tag "joined"
        valueToStore (get global joined)
      call TinyDB1 .StoreValue
        tag "userName"
        valueToStore (get global userName)
      set Label1 . Text to "Type in your message and press Submit to send..."
```



# Lokální uložení jména

```
when Screen1.Initialize
do
  set global joined to call TinyDB1.GetValue
    tag "joined"
    valueIfTagNotThere false
  if get global joined
  then
    set Label1.Text to "Type in your message and press Submit to send..."
    set global userName to call TinyDB1.GetValue
      tag "userName"
      valueIfTagNotThere ""
  call CloudDB1.GetValue
    tag "message"
    valueIfTagNotThere create empty list
```

# Vytvoření místností

- Je třeba v Designer módu přidat User Interface -> Spinner s vyplněnými Elements

```
initialize global room to " MUNI "
```

```
when Spinner1 .AfterSelecting  
  selection  
do  
  set global room to get selection  
  call CloudDB1 .GetValue  
    tag get global room  
    valueIfTagNotThere create empty list
```

The image shows a sequence of code blocks in a visual programming language. The first block is an orange 'initialize global' block with 'room' as the variable and 'MUNI' as the value. Below it is a yellow 'when' block for 'Spinner1' with the event 'AfterSelecting'. Inside this 'when' block is a 'do' loop. The first block in the 'do' loop is an orange 'set global' block for 'room' with the value 'get selection'. The second block is a purple 'call' block for 'CloudDB1' with the method '.GetValue'. It has two arguments: 'tag' with the value 'get global room' and 'valueIfTagNotThere' with the value 'create empty list'.

# Vytvoření místností

- Je třeba nahradit všechny tagy “message” proměnnou room

The image displays two blocks of Scratch code. The top block is triggered by 'when CloudDB1 .GotValue' and contains a 'do' loop with an 'if' condition. The 'if' condition checks 'is in list? thing' with 'get tag' as the 'thing' and 'Spinner1 . Elements' as the 'list'. If true, it executes two 'set' blocks: 'set global chatList to get value' and 'set ListView1 . Elements to get global chatList'. The bottom block is triggered by 'when CloudDB1 .DataChanged' and contains an identical 'do' loop with the same 'if' condition and 'set' blocks.

## Další možnosti vylepšení:

- Tlačítko Clear, které vymaže všechny zprávy
- Tlačítko Change name, kterým uživatel může změnit své jméno
- Automatické zaslání zprávy, že někdo vstoupil do konverzace (např. “\*Alice has joined the chat.\*”)
- Zobrazování nových zpráv na vrcholu seznamu
- ...

# CO NÁS ČEKÁ PŘÍŠTĚ

# 7. Práce s daty

- Využití dat
- Velká data
- Databáze, cloud computing
- Správa dat, data science
- Digitální stopa uživatele, etické kontroverze