

## 3.2. Dychový tréner

### Kľúčové slová

vizualizácia, animácia, zvuk, viac obrazoviek, hodiny, časovač, prepínač, farby, zoznamy.

### Čo sa naučíme a čo si precvičíme

- poskytovať informácie v rôznej forme – text, tabuľka, infografika, animácia, zvuk,
- navrhnuť farebnú paletu aplikácie,
- pracovať s dvomi obrazovkami (komponenty `Screen`),
- používať zoznamy údajov,
- kresliť na plátno (komponent `Canvas`),
- používať komponent `Clock` na generovanie udalostí v pravidelnom časovom intervale,
- animovať pohyb guľôčky (komponent `Ball`),
- použiť komponenty `CheckBox` (zaškrŕavacie políčka) ako `Radio Buttons` (prepínače),
- generovať hlasové a zvukové výstupy (komponenty `TextToSpeech`, `Sound`).

### Príprava na výučbu

Prerekvizity: animovanie pohybu guľôčky, časovač (etuda 2.3), práca s dvomi obrazovkami (etuda 2.5), spracovanie zoznamov (etuda 2.8).

Pomôcky:

Elektronický pracovný list `pmz_3_2_Dychovy trener_PL.xlsx` obsahuje tabuľky na doplnenie údajov odčítaním z grafu. Doplnením vzorcov sa dá experimentovať s nastavovaním rôznych parametrov tréningu. Pracovný list sa môže aj vytlačiť, ale potom je statický a experimentovať s prepočítavaním údajov sa nedá. Práca s pracovným listom slúži na dôkladné pochopenie problému pred začiatkom programovania aplikácie.

Aplikácia `pmz_3_2_RadioCheck.aia` na preskúmanie.

### Čo zaujímavé sa môžeme dozvedieť?

Nádychové potápanie (free-diving) je potápanie bez dýchacieho prístroja len so zásobou vzduchu v pľúcach. Oblúbené je rekreačné potápanie na hladine mora s hlavou pod vodou so šnorchlom na dýchanie a s príležitostným ponorením do hĺbky, pri ktorom treba zadržať dych.

Nádychové potápanie je aj športová disciplína v rôznych súťažných kategóriách:

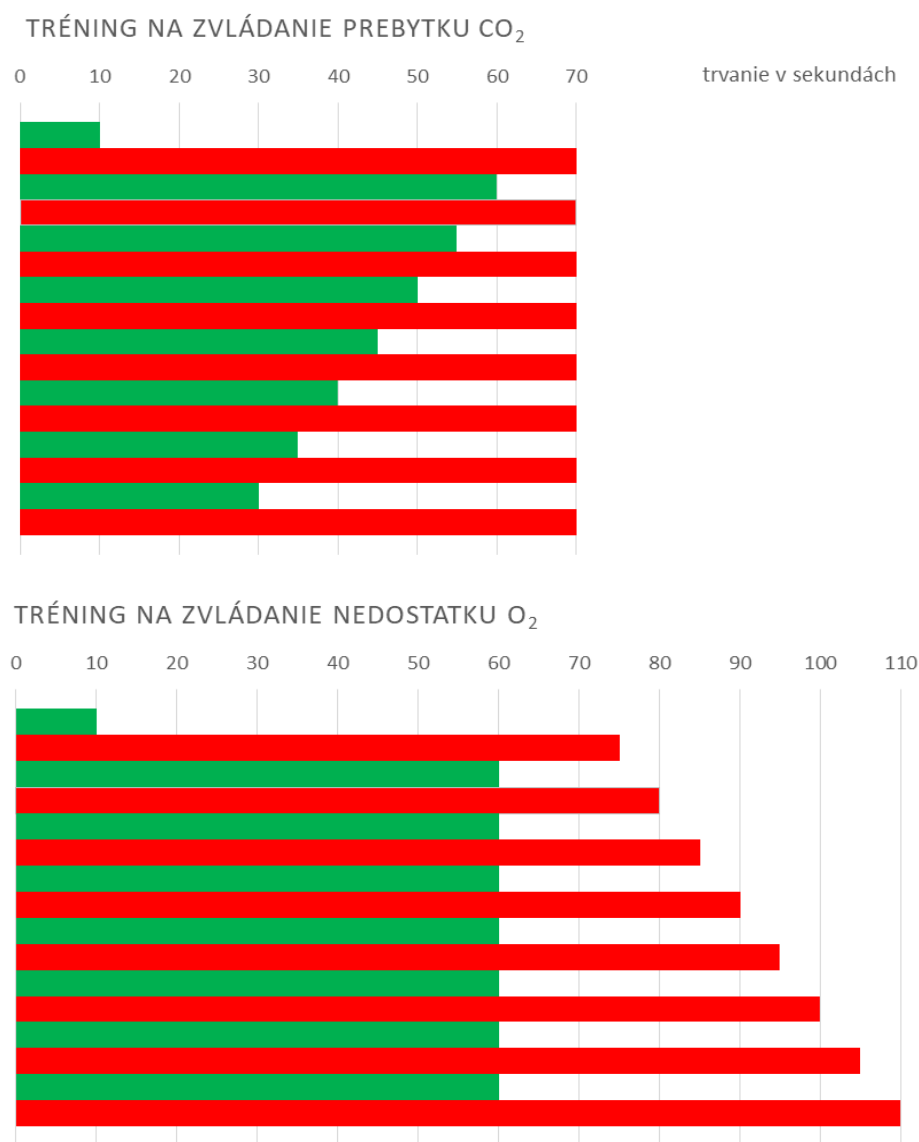
- statická apnea – zadržanie dychu na čas v plytkej vode (v bazéne),
- dynamická apnea – plávanie pod vodou na vzdialenosť (s plutvami alebo bez plutiev),
- hĺbkové potápanie – dosahovanie čo najväčšej hĺbky (so závažím, bez závažia, s plutvami, bez plutiev, s lanom, bez lana).

Viac o nádychovom potápaní a o rekordoch v tomto športe nájdete vo wikipédii [https://sk.wikipedia.org/wiki/Nádychové\\_potápanie](https://sk.wikipedia.org/wiki/Nádychové_potápanie).

Zlepšiť výkonnosť v dĺžke zadržania dychu sa dá tréningom. Pri zadržaní dychu ľudský organizmus bojuje s dvomi problémami:

- nedostatok kyslíka ( $O_2$ ) v krvi,
- prebytok oxidu uhličitého ( $CO_2$ ) v krvi.

Zvýšiť odolnosť organizmu voči nedostatku kyslíka v krvi sa dá tréningom, pri ktorom sa postupne predlžuje interval zadržania dychu pri zachovaní rovnakej dĺžky odpočinku až do 80 % hodnoty osobného maxima. Ak chceme trénovať odolnosť voči prebytku oxidu uhličitého v krvi, budeme postupne znižovať čas odpočinku pri zachovaní dĺžky zadržania dychu, ktorá by mala byť 50 % hodnoty osobného maxima.



Obr. 3.2.1 Grafy: Tréningové plány na zvládanie prebytku oxidu uhličitého a nedostatku kyslíka v krvi pri zadržaní dychu

Na obrázku 3.2.1 je sú grafy znázorňujúce tréningové plány na zvládanie prebytku CO<sub>2</sub> a nedostatku O<sub>2</sub> v krvi. Zelené sú časové intervaly dýchania, červené zadržania dychu.

### Úloha 1

Vyplňte elektronický pracovný list podľa grafov na obrázku 3.2.1:

- Údaje z grafu zaznačte do tabuľky.
- Zistite, aké je osobné maximum zadržania dychu osoby, pre ktorú je určený tréningový plán.
- Určte parametre tréningov:
  - úvodné rozdýchanie,
  - začiatočná dĺžka zadržania dychu,
  - zmena dĺžky zadržania dychu,
  - začiatočná dĺžka vydýchania,
  - zmena dĺžky vydýchania,
  - počet opakovaní.
- Zostavte vzorce, ktoré podľa parametrov tréningu dynamicky počítajú tréningové časy dýchania a zadržania dychu. Experimentujte s rôznymi parametrami tréningu.

### Akú zaujímavú aplikáciu môžeme vytvoriť?

Mobilné zariadenia môžu slúžiť športovcom na plánovanie alebo zaznamenávanie priebehu tréningov.

#### Otázky na zamyslenie

Poznáte nejaké aplikácie na podporu športových aktivít? Aké služby poskytujú?

Vytvoríme aplikáciu, ktorá bude asistovať svojmu používateľovi pri tréningu na zvýšenie výkonnosti v dĺžke zadržania dychu. V pracovnom liste sme už vytvorili tabuľku, ktorá počíta časové hodnoty jednotlivých úsekov tréningu. Mobilná aplikácia, ktorú vytvoríme, bude poskytovať používateľovi ďalšie užitočné služby. Číselné údaje v tabuľke názornejšie zobrazí vo forme infografiky – grafického zobrazenia dát, ktoré doplníme ešte animáciou zobrazujúcou priebeh tréningu. Mobilné zariadenie použijeme ako stopky na meranie času a okrem vizuálneho zobrazenia priebehu tréningu pridáme aj zvukové upozornenia v dôležitých úsekoch tréningu. Naša aplikácia bude:

- umožňovať nastavovanie parametrov tréningu,
- zobrazovať tréningový plán vo forme grafu,
- asistovať pri tréningu meraním času,
- zvukovo upozorňovať používateľa na zmenu akcie (dýchanie, zadržanie dychu),
- animovať priebeh tréningu pohybom kurzora po grafickom pláne.

### Ako budeme postupovať pri tvorbe aplikácie?

#### Úloha 2

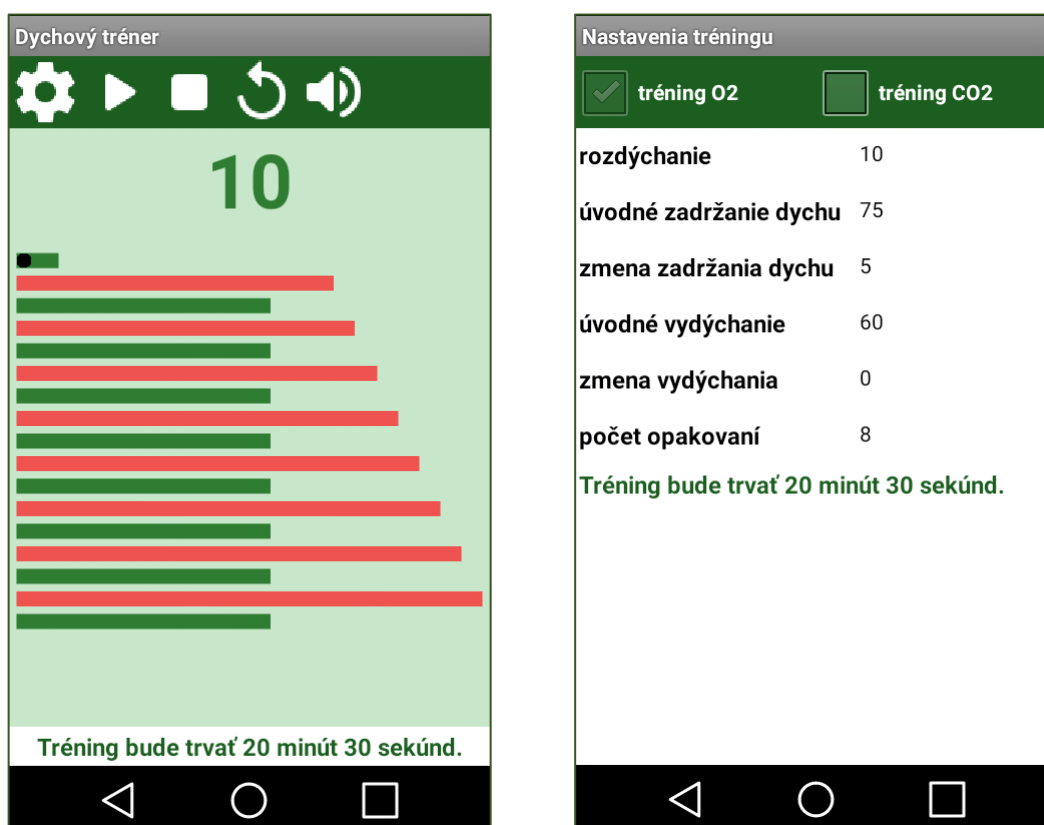
Vytvorte používateľské rozhranie aplikácie.

Vytvoríme dizajn aplikácie s dvomi obrazovkami napríklad ako na obr. 3.2.2. Na prvú obrazovku vložíme:

- pás (komponent `HorizontalArrangement`) ovládacích tlačidiel (komponenty `Button`) na nastavenie parametrov tréningu, spustenie tréningu, zastavenie tréningu, resetovanie tréningu, zapnutie/vypnutie zvuku,
- nápis (komponent `Label`) s odpočítavaním sekúnd počas tréningu,
- oblasť obrazovky (komponent `Canvas`), do ktorej nakreslíme tréningový plán podľa nastavených parametrov,
- kurzor (komponent `Ball`) označujúci miesto, kde sa nachádzame v pláne počas tréningu.

Na druhú obrazovku vložíme:

- dva komponenty `CheckBox` na voľbu typu tréningu (O2 alebo CO2),
- tabuľku (komponent `TableArrangement`) s názvami parametrov tréningu (komponenty `Label`) a ich hodnotami (komponenty `TextBox`).



Obr. 3.2.2 Vzhľad aplikácie: Obrazovka s priebehom tréningu, obrazovka na nastavovanie parametrov tréningu

Aby naša aplikácia mala pekný dizajn, vyberme vhodné farby a pre tlačidlá vhodné piktogramy. Vzájomne ladiace odtiene farieb a rôzne piktogramy nájdeme napríklad na stránke <https://www.materialpalette.com/>.

### Vysvetlíme si

Blok `make color` zo skupiny vstavaných blokov *Colors* vytvára farbu zmiešaním trojice farieb (odtiev červené (Red), zelenej (Green) a modrej (Blue)). Odtiene farieb sa udávajú ako čísla veľkosti 1 bajt v desiatkovej sústave (0-255) vložené do trojprvkového zoznamu. Napríklad:



Kódy farieb sa často zvyknú udávať v hexadecimálnom (šestnástkovom) kóde. Napríklad:

`#1b5e20` je tmavozelená farba, kde 1b, 5e, 20 sú odtiene farieb RGB, ktoré pre použitie v bloku `make color` musíme previesť do desiatkovej sústavy na 27, 94, 32.

Správanie sa aplikácie začneme programovať grafickým zobrazením tréningového plánu.

### Úloha 3

Naprogramujte funkciu `kresliGraf`, ktorá do komponentu `Canvas` vykreslí striedavo zelené a červené pruhy, ktorých dĺžky sú dané v globálnej premennej `rozvrh` typu zoznam (list).

Do globálnej premennej `rozvrh` si na testovacie účely vložme zoznam s časovými hodnotami úsekov tréningu z pracovného listu (10, 75, 60, 80, 60, 85, 60, 90, 60, 95, 60, 100, 60, 105, 60, 110). Každé číslo zo zoznamu vizualizujeme ako vodorovnú čiaru, striedavo zelenou a červenou farbou (začneme zelenou farbou).

### Vysvetlíme si

Vlastnosti komponentu `Canvas`:

`PaintColor` – farba, ktorou sa kreslí

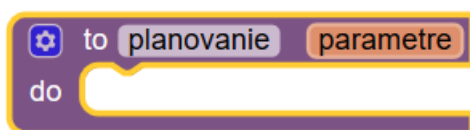
`LineWidth` – hrúbka čiary

Metóda komponentu `Canvas`:

`DrawLine(x1, y1, x2, y2)` – kreslí úsečku nastavenej hrúbky a farby s krajnými bodmi na súradniciach (x1, y1) a (x2, y2)

### Úloha 4

Naprogramujte funkciu `planovanie`, ktorá pre zadaný vstupný zoznam `parametre` tréningu vygeneruje zoznam časových úsekov tréningu do globálnej premennej `rozvrh`.



`parametre` je zoznam parametrov tréningu podľa tabuľky v pracovnom liste: úvodné rozdychanie, začiatočná dĺžka zadržania dychu, zmena dĺžky zadržania dychu, začiatočná dĺžka vydýchania, zmena dĺžky vydýchania, počet opakovaní.

Na testovacie účely nastavme zoznam `parametre` podľa údajov v pracovnom liste na (10, 75, 5, 60, 0, 8). Funkcia `planovanie` pre takéto `parametre` vygeneruje zoznam `rozvrh` = (10, 75, 60, 80, 60, 85, 60, 90, 60, 95, 60, 100, 60, 105, 60, 110)

Vygenerovaný rozvrh tréningu zobrazíme pomocou funkcie `kresliGraf`. Experimentujte s rôznymi parametrami tréningu a kontrolujte grafický výstup s hodnotami, ktoré vygenerujú vzorce v tabuľke pracovného listu.

Pri experimentovaní s rôznymi parametrami tréningu sa môže stať, že vygenerované časové hodnoty jednotlivých úsekov tréningu budú príliš veľké alebo príliš malé pre zobrazenie na displeji mobilného zariadenia. Problém s grafickým zobrazením údajov vo vhodnej mierke vyriešime naprogramovaním špeciálnej funkcie, ktorá bude číselný údaj v sekundách prevádzať na vhodný údaj v pixeloch, ktorý sa prispôsobí veľkosti displeja zariadenia.

### Úloha 5

Naprogramujte funkciu `mierka`, ktorá pre zadané číslo (čas z tréningového rozvrhu) vypočíta dĺžku úsečky tak, aby sa maximálny časový úsek z tréningu zobrazil cez celú obrazovku.

Funkciu `mierka` použijeme vo funkcii `kresliGraf` pri vizualizácii tréningového plánu.

#### Vysvetlíme si

Mierka je pomer šírky plátna (`Canvas.Width`) a maximálneho času v tréningovom pláne (zoznam časových hodnôt v globálnej premennej `rozvrh`). Príklad výpočtu mierky a jej použitia pre výpočet dĺžky úsečky:



```
rozvrh = (10, 75, 60, 80, 60, 85, 60, 90, 60, 95, 60, 100, 60, 105, 60, 110)
max. čas v zozname rozvrh = 110
šírka plátna Canvas.Width = 320
mierka = 320/110
dĺžka čiary = čas * mierka
dĺžka čiary pre 110 sekúnd = 110 * 320/110 = 320
dĺžka čiary pre 10 sekúnd = 10 * 320/110 = 29
```

**Poznámka:** Maximálny čas v tréningovom pláne (zoznam `rozvrh`) treba zistiť výpočtom.

Teraz pridajme do nášho projektu interaktivitu. Parametre tréningu nebudú v aplikácii nastavené napevno, ale budú sa dať nastavovať a meniť priamo v aplikácii.

### Úloha 6

Naprogramujte interaktívne nastavovanie parametrov tréningu pre funkciu `planovanie` na samostatnej obrazovke aplikácie.

Na druhú obrazovku aplikácie prejdeme stlačením tlačidla (udalosť `Button_Nastavenia.Click`). Po nastavení parametrov tréningu v editovacích poliach sa vrátíme späť na prvú obrazovku (systémovým tlačidlom Back). Pri návrate na hlavnú obrazovku aplikácie (udalosť `Screen1.OtherScreenClosed`) odovzdáme zoznam parametrov tréningu ako výsledok v premennej `result` a spracujeme ho: vytvoríme rozvrh tréningu volaním funkcie `planovanie` a vizualizujeme ho volaním funkcie `kresliGraf`.

### Úloha 7

Naprogramujte odpočítavanie času podľa tréningového plánu uloženého v globálnej premennej `rozvrh`:

- textové zobrazenie počtu zostávajúcich sekúnd daného úseku tréningu, farebne odlišť text pri dýchaní zelenou a pri zadržaní dychu červenou farbou,
- hlasové upozornenie pri zmene: „Dýchaj“ alebo „Zadrž dych“,
- zvukové upozornenie pred zmenou: 5 sekúnd pred zmenou pridajte zvukové pípanie,
- animáciu pohybu kurzora (guľôčky) po grafickom zobrazení rozvrhu tréningu.

Ako časovač na odpočítavanie času v sekundách využijeme komponent `Clock` a jeho schopnosť generovať udalosti v pravidelných časových intervaloch. Necháme každú sekundu vygenerovať udalosť, na ktorú bude aplikácia reagovať vypísaním zostávajúceho času, posunom kurzora v grafickom pláne, za istých podmienok zvukovým signálom alebo hlasovým upozornením. Na textové zobrazenie odpočítavania sa hodí komponent `Label`. Na hlasové upozornenia pri zmene úseku tréningu využijeme komponent `TextToSpeech` na prevod textu do hovorenej reči. Zvukové pípanie pred koncom každého časového úseku tréningu realizujeme s využitím komponentu `Sound`. Ako pohybujúci sa kurzor na animovanie priebehu tréningu po grafickom pláne môžeme použiť komponent `Ball`.

V etudách v kapitole 2 sme používali prvky, ktoré môžeme použiť aj v tomto projekte na realizáciu odpočítavania času podľa časového rozvrhu tréningu:

Komponenty:

- `Clock`
- `Label`
- `TextToSpeech`
- `Sound`
- `Ball`

Vlastnosti:

- `Clock.TimerEnabled`, `Clock.TimerAlwaysFires`, `Clock.Interval`
- `Label.TextColor`, `Label.Text`
- `Sound.Source`

Udalosť:

- Clock.Timer

Metódy:

- TextToSpeech.Speak
- Sound.Play
- Ball.MoveTo

### Úloha 8

Naprogramujte nastavovanie predvolených parametrov dvoch typov tréningu (na zvládanie nedostatku O<sub>2</sub> a na zvládanie prebytku CO<sub>2</sub> v krvi) pomocou začiarkavacích políčk na druhej obrazovke aplikácie.

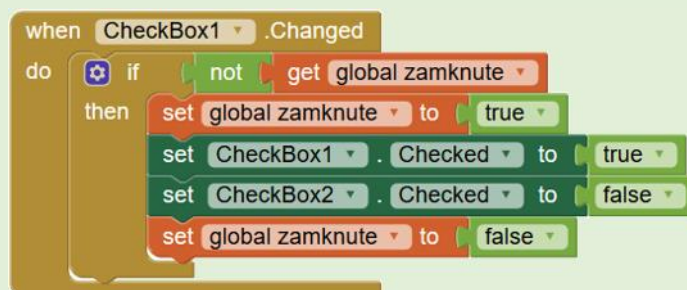
#### Vysvetlíme si

*Check Box* alebo začiarkavacie políčko je grafický ovládací prvok, pomocou ktorého sa nastavuje (začiarkne/odčiarkne) logická hodnota (áno/nie) nejakej premennej na opačnú.

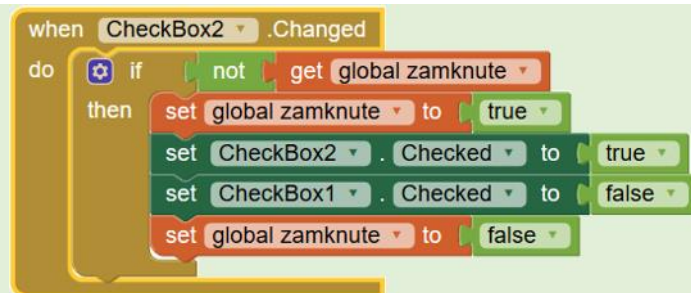
*Radio Button* alebo tlačidlo možností je grafický ovládací prvok, pomocou ktorého sa vyberá jedna možnosť zo skupiny možností (nastavuje sa logická hodnota *áno* práve jednému zo skupiny tlačidiel, ostatné sa nastavujú na *nie*).

Komponent `CheckBox` v App Inventore poskytuje funkčnosť začiarkavacieho políčka. Pre tlačidlo možností (*Radio Button*) App Inventor nemá vstavaný komponent, jeho funkčnosť môžeme naprogramovať pomocou políčk `Check Box`:

Dotykom na políčko `CheckBox` (udalosť `Changed`) nastavíme jeho stav vždy na `true` a stav ostatných políčk `CheckBox` v skupine na `false`. Táto zmena stavu ostatných políčk však tiež vyvolá udalosť `Changed` a kód sa vykoná aj pre ne. Aby sme tomu zabránili, podmienime reagovanie na túto udalosť pomocou globálnej logickej premennej, ktorou dočasne „uzamkneme“ reakcie na zmeny, až kým všetky políčka nebudú nastavené na správnu hodnotu. Potom opäť „odomkneme“ vykonávanie reakcií na udalosť `Changed` komponentov `CheckBox`. Riešenie pre výber z dvoch možností:







Poznámka: Všeobecné riešenie pre ľubovoľný počet tlačidiel možností v skupine preskúmajte v projekte pmz\_3\_2\_RadioCheck.aia.

### Ako vylepšiť či rozšíriť našu aplikáciu?

Dobrá aplikácia by mala mať tieto vlastnosti:

- *funkčnosť* – mala by byť dobre použiteľná na to, na čo je určená,
- *efektívnosť* – nemala by vyžadovať od používateľa príliš veľké úsilie na dosiahnutie cieľa (pri prvom kontakte s aplikáciou, ani pri dlhodobejšom používaní alebo po návrate k aplikácii po dlhšom čase),
- *odolnosť voči chybám používateľa* – mala by predchádzať chybám používateľa a ak k nim dôjde, účinne ich riešiť,
- *príjemnosť* – práca s aplikáciou by mala vyvolávať u používateľa spokojnosť, príjemné pocity.

### Otázky na zamyslenie

Sú služby, ktoré aplikácia poskytuje, užitočné pre realizáciu dychového tréningu? Mohla by mať nejaké ďalšie užitočné funkcie?

Vie s aplikáciou jednoducho pracovať človek pri prvom kontakte? Je ovládanie aplikácie efektívne aj pre skúseneho používateľa? Mohlo by sa ovládanie aplikácie nejako zjednodušiť, vylepšiť?

Je aplikácia odolná voči chybám používateľa? Rieši problémy s chybnými vstupmi?

Má aplikácia príjemný dizajn (farby, grafiku, zvuky)?

Niekoľko námetov na vylepšenie aplikácie:

- užitočná funkcia: vypočítanie a zobrazenie celkovej dĺžky tréningu,
- pomôcky pre zadávanie vstupov, napr. nápovede, z akého intervalu vyberať vstupy pre zvolený typ tréningu, informačné texty o dychovom tréningu,
- predchádzanie chybným vstupom, riešenie chybných vstupov,
- zladenie farebnej palety, starostlivý výber zvukových efektov, piktogramov.

### Čo sme sa naučili

- ✓ vytvárať aritmetické postupnosti údajov podľa vstupných parametrov,
- ✓ programovať parametrizovanú grafiku, miešať farby,

- ✓ pracovať s dvomi obrazovkami, prenášať údaje medzi nimi,
- ✓ naprogramovať funkcionality skupiny tlačidiel možností pomocou začiarovacích políček,
- ✓ programovať zvukové výstupy s využitím časovača.

Odporúčaný priebeh výučby		
Činnosť učiteľa	Činnosť žiaka	Poznámky
<b>1. hodina – Úvod, úlohy 1 a 2</b>		
Učiteľ vedie motivačný výklad o nádychovom potápaní a dychovom tréningu.	Žiaci sledujú výklad učiteľa, vyhľadávajú ďalšie zaujímavé informácie o nádychovom potápaní vo wikipédii.	Je dôležité vzbudiť záujem žiakov o tému projektu, preto venujeme dostatočný čas na samostatné hľadanie zaujímavých informácií na webe (napr. rekordy v nádychovom potápaní).
Učiteľ zadá problémovú úlohu 1 – vyčítať informácie z grafov.	Žiaci riešia úlohu 1 v e-pracovnom liste.	Cieľom úlohy 1 je dôkladne sa oboznámiť s princípmi dychového tréningu s konkrétnymi hodnotami a so zovšeobecnením pomocou parametrov.
Učiteľ položí otázku na zamyslenie o aplikáciách na asistovanie športových tréningov. Demonštruje vzorovú aplikáciu a zadá úlohu 2.	Žiaci diskutujú o aplikáciách pre športovcov. Navrhnu hrubý dizajn aplikácie Dychový tréner (riešenie úlohy 2).	Dizajn aplikácie nemusí byť totožný s dizajnom prezentovaným učiteľom, ale má obsahovať všetky požadované prvky podľa zadania.
<b>2. hodina – úlohy 3 až 5</b>		
Úlohy 3, 4: grafické zobrazenie tréningového plánu. Učiteľ usmerňuje prácu žiakov.	Žiaci programujú grafické znázornenie tréningu. Aplikáciu testujú v živom ladení s napevno zadanými hodnotami vstupov podľa údajov z pracovného listu.	Metodika je zvolená tak, aby žiaci postupovali od konkrétneho k všeobecnému (kreslenie podľa konštantného rozvrhu, generovanie rozvrhu podľa konštantných parametrov z pracovného listu).
Úlohy 5: Učiteľ v problémovom výklade vysvetlí princíp mierky.	Žiaci s pomocou učiteľa zostavujú funkciu na výpočet dĺžky čiary v grafickom znázornení časového úseku tréningu. Testujú správnosť vykresľovania pre rôzne parametre tréningu v živom ladení.	Výsledkom práce je grafické znázornenie tréningu na šírku obrazovky pre časový rozvrh generovaný z parametrov tréningu (zadaných napevno).
<b>3. hodina – úlohy 6 a 7</b>		

Úloha 6: Učiteľ zopakuje používanie komponentu Clock na generovanie udalostí v pravidelných časových intervaloch. Demonštruje číselné, zvukové, animované odpočítavanie času na vzorovej aplikácii. V prípade potreby pomáha žiakom v programovaní.	Žiaci samostatne programujú multimediálne odpočítavanie času. Priebežne si navzájom zdieľajú svoje nápady, riešenia.	Žiaci pracujú samostatne, ale podporuje sa ich vzájomná komunikácia a spolupráca.
Úloha 7: Učiteľ zopakuje spôsob odovzdávania údajov medzi dvomi obrazovkami.	Žiaci si pripomenú spôsob práce s dvomi obrazovkami. Testujú zadávanie rôznych parametrov tréningu na druhej obrazovke.	Odporúčame využiť e-pracovný list na kontrolu správnosti generovania údajov a ich vykresľovania.
<b>4. hodina – úloha 8, rozširujúce námety</b>		
Učiteľ vysvetlí spôsob realizácie ovládacích prvkov Radio Button pomocou ovládacích prvkov CheckBox.	Žiaci podľa vzoru učiteľa naprogramujú výber typu tréningu pomocou CheckBoxov s funkcionalitou Radio Button.	Žiaci si majú uvedomiť rozdiel medzi dvomi typmi ovládacích prvkov a obmedzenia App Inventora v ponuke ovládacích prvkov.
Učiteľ navodí diskusiu o kritériách použiteľnosti softvérových aplikácií. Vyzve žiakov na generovanie nápadov na vylepšenie aplikácie.	Žiaci diskutujú a prispievajú nápadmi do spoločnej diskusie. Pracujú na vylepšeniach svojich aplikácií, dokončujú projekt.	Cieľom diskusie je sformulovať vlastnosti dobre použiteľného softvéru a na základe nich navrhnúť vylepšenia vlastných aplikácií.
<b>5. hodina – testovanie, prezentovanie, hodnotenie projektov</b>		
Učiteľ organizuje vzájomné testovanie aplikácií žiakmi a záverečné rovesnícke hodnotenie projektov. Ohodnotí projekty.	Žiaci si navzájom vymenia a testujú svoje aplikácie. V záverečnej diskusii sa vyjadria ku každej aplikácii v kritériách funkčnosť, efektívnosť, odolnosť, príjemnosť.	Záverečné hodnotenie v určenom čase je vyvrcholením práce na projekte. Hodnotenie musí byť štruktúrované podľa jasných kritérií.

## Bibliografia

Michaličková, V. (2016). *Programovanie mobilných aplikácií v prostredí MIT App Inventor 2*. Nitra: Univerzita Konštatína Filozofa v Nitre.

MIT. (2018). . Retrieved from: <http://ai2.appinventor.mit.edu/>

Wolber, D., Abelson, H., Spertus, E., & Liz, L. (2014). *App Inventor 2 - Create Your Own Android Apps*. O'Reilly.

<https://www.yachtmeni.cz/freediving-apnea/apnea-trenink/>

<https://www.materialpalette.com/colors>

<https://www.materialpalette.com/icons>

<http://www.ignas.in/tutorial/radiobutton>

<https://cs.wikipedia.org/wiki/Použitelnost#Definice>

## Register pojmov

### C

Canvas.DrawLine, 5  
Canvas.LineWidth, 5  
Canvas.PaintColor, 5

### I

infografika, 2

### M

make color, 5

### R

RGB, 5