**45 min. do 90 min. 🡪 BEEP**

(robotek je ves čas priključen na kabel, ne potrebuje baterije)

Učencem povem, da se vsak ukaz lahko zapiše na dva načina – en način je zapis v SET UP, kar pomeni, da se bo ukaz izvedel samo enkrat, drug način je zapis v LOOP, kar pomeni, da se bo ukaz ponavljal v nedogled, dokler ga sami ne prekinemo.

Vstavimo blokec:

Set up

Loop

Razložimo ga s pomočjo funkcije beep.

***1. del*** V SET UP vstavijo beep:

Beep (ton=C, t=1) oboje je že vpisano notri (lahko pa spremenijo) (enota t[sekunda])

* Robota s kablom povežeš na računalnik.
* V programu Ardublockly (na računalniku) oblikuješ ukaz in ga naložiš na robota (program pošlje sporočilo po kablu na robota).
* Takoj se izvede ukaz, torej robot enkrat zapiska, nato pa preneha.

Robot je v tem primeru zapiskal samo enkrat, saj smo funkcijo beep zapisali v del programa SET UP (ta omogoča, da se ukaz izvede le enkrat).

***2. del***  Če želimo, da robot ponavlja določen ton neskončno dolgo, pa BEEP funkcijo vstavimo v VOID LOOP (ta omogoča, da se ukaz ponavlja v neskončnost).

* Okvirčke iz zgornje SET UP zanke primeš in preneseš v spodnjo LOOP zanko.
* Ponovimo postopek nalaganja programa (ukaza) na robota od prej:
  + Robota s kablom povežeš na računalnik.
  + Ukaz naložiš na robota.
  + Takoj se izvede ukaz, torej robot piska, piska, piska … saj je ukaz zapisan v VOID LOOP, ki je neskončna zanka in ukaz ponavlja neštetokrat. ZATO je potrebno kabel iz računalnika (~~robota~~, zaradi možnosti poškodb) iztakniti in s tem robotu preprečiti nadaljnje delovanje. Tako robot nima več energije za delovanje, kar pomeni , da se izklopi/ugasne in hkrati preneha s piskanjem.
* OPOZORILO: ob ponovni priključitvi robota na kabel oz. računalnik bo robot začel piskati (dokler nanj ne naložimo novega programa/ukazov).

***3. del*** Sedaj sestavimo melodijo, torej več BEEP funkcij povežemo eno za drugo.

**BEEP**

**BEEP**

**BEEP**

**DELAY**

**DELAY**

* En BEEP blokec je že v LOOPU, dodamo še dva:
  + **Pod** že obstoječi blokec, pripneš nov, ENAK blokec (stojita en pod drugim, ukaz se izvaja en za drugim).
  + Dodaš lahko še blokec funkcije DELAY (počakaj).
  + Zlagaš blokce enega pod drugim:

**4. del Samo demonstracija učitelja**

* Lahko bi pustili vse frekvence enake (robot vedno piska z istim tonom), lahko pa so piski drugačni. Da melodija ne bo dolgočasna, na enem BEEP blokcu spremenimo frekvenco oz. izberemo drug ton.
* Učitelj naloži program na robota in ta piska v neskončnost. Ker nam to ni všeč, uvedemo IF ZANKO.

Učitelj hkrati nazorno kaže na projekciji potek ukazov.

**5. del** RAZLAGA IF ZANKE: Robotu postaviš pogoj, npr: **če** **pritisneš gumb na robotu, naj se piskanje prične.** Dokler gumba na robotu ne pritisneš, se ukazi (blokci) ne bodo začeli izvajati. Ko pritisneš gumb, se izvedejo vsi ukazi v LOOP zanki (vsi blokci beepa – 1., 2., 3., 4., 5., nato se vrne nazaj na začetek LOOP in se ne zgodi nič. Če želimo melodijo ponovno vklopiti, moramo zopet pritisniti gumb – ker smo robotu postavili pogoj.

* Vstaviš blokec IF.
* Naš pogoj bo pritisk gumba, zato vstaviš še blokec, na katerem piše PRITISK GUMBA.

**BEEP**

**BEEP**

**BEEP**

**DELAY**

**DELAY**

**IF – PRITISK GUMBA**

* Vsak učenec nastavi blokce, ukaz naloži na robota in preizkusi.

🡪 Morebitna VPRAŠANJA

**6. del** UČNI LIST

Vsak učencev ima vse naloge na listu, sam izbere katere naloge bo reševal.

Tabela tonov s pripadajočimi frekvencami za vsak ton. (6. vrstica - <https://www.seventhstring.com/resources/notefrequencies.html>)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ton** | C | D | E | F | G | A | H | c' |
| **frekvenca** |  |  |  |  |  |  |  |  |

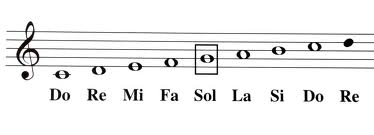
1. OSNOVNI NIVO

Vstavijo blokce beep, izberejo izmed podanih tonov in ne spreminjajo trajanja tonov (enako trajanje tonov čez celo pesem, npr. v četrtinkah).

1. SREDNJI NIVO

Vstavijo blokce beep, v katere vstavijo frekvence tonov, in ne spreminjajo trajanja tonov (enako trajanje tonov čez celo pesem, npr. v četrtinkah).

1. VIŠJI NIVO

Vstavijo blokce beep, v katere vpišejo frekvence in IN trajanje tonov (npr. 1 celinka  = 1 sekunda)

