

PROGR@MUJ W ZESPOLE MATERIAŁY DLA NAUCZYCIELI



Progr@muj w zespole

PROGR@MUJ W ZESPOLE MATERIAŁY DLA NAUCZYCIELI

ADAM JURKIEWICZ
RAFAŁ KAMIŃSKI
KONRAD KOSIERADZKI
ELŻBIETA PIOTROWSKA-GROMNIAK

2022

Innowacja społeczna **„Progr@muj w zespole”**
została zrealizowana dzięki wsparciu uzyskanemu
w ramach projektu „POPOJUTRZE 2.0 – KSZTAŁCENIE”
nr POWR.04.01.00-00-I108/19
współfinansowanego ze środków Unii Europejskiej
w ramach Europejskiego Funduszu Społecznego
oraz środków budżetu państwa

Realizacja projektu:

Liceum Ogólnokształcące z Oddziałami Dwujęzycznymi
im. Adama Mickiewicza w Piastowie

Autorzy:

Adam Jurkiewicz
Rafał Kamiński
Konrad Kosieradzki
Elżbieta Piotrowska-Gromniak

Redakcja i korekta:

Maria Gudro-Homicka

Projekt graficzny i skład:

Zuzanna Piwońska

Konsultacje i wsparcie:

Marcin Bielicki
Anna Kusiak
Mariusz Mażewski
Agata Pakieła
Łukasz Wolff

MATERIAŁY DLA NAUCZYCIELI – SPIS TREŚCI

<i>Od autorów</i>	6
<i>Podziękowania</i>	9
<i>Wstęp</i>	11
<i>Przewodnik dla nauczycieli</i>	16
Wprowadzenie	16
Zespoły	22
Zasady pracy – kontrakt	26
Role w zespole	30
Konflikt	31
Słów kilka o komunikowaniu się	35
Gdy pojawiają się emocje...	37
<i>Badanie umiejętności współpracy i komunikacji w zespole</i>	40
Wprowadzenie	40
Struktura testu	40
Zastosowanie testu	41
Założenia organizacyjne	41
<i>Test umiejętności współpracy i komunikacji w zespole</i>	43
Arkusze z kluczem punktacji	58
Ogólne wskazówki i sugestie dla nauczyciela	58

Sposób obliczania punktacji dla pięciu obszarów	59
Scenariusze poszczególnych zajęć	73
Zajęcia 0: Wprowadzenie w projekt	74
Zajęcia 1: Przygotowanie środowiska – konta, oprogramowanie	79
Zajęcia 2: Virtualbox i pozostałe aplikacje	85
Zajęcia 3: Python – podstawy, dane, pętle, instrukcje warunkowe	89
Zajęcia 4: Podstawy edytorów i pętla while w Python	96
Zajęcia 5: Elementy sterujące w PySimpleGUI	100
Zajęcia 6: Dalsze działania w edytorach, moduł requests w Python Console	104
Zajęcia 7: Format JSON, pętle, listy, słowniki w Python	107
Zajęcia 8: Edytory, dokumentacja i API_KEY dla przykładowych projektów	111
Zajęcia 9: Funkcje, zasięg zmiennych – NAMESPACE, własne repozytorium	115
Zajęcia 10: Testy dostępu do API oraz współpraca z GitHub	119
Zajęcia 11: Sprawdzanie połączeń do API, manipulowanie elementami słowników	124
Zajęcia 12: Prezentacja rezultatów pracy zespołów i podsumowanie projektu	128



OD AUTORÓW

Dlaczego i dla kogo przygotowaliśmy innowację społeczną:
„Progr@muj w zespole”?

Od wielu lat, biorąc udział w różnych konferencjach, warsztatach czy choćby w rozmowach z uczniami i nauczycielami, słyszeliśmy o fascynacji młodzieży programowaniem. Małe dzieci często mówią np. o Scratchu i traktują programowanie jako zabawę, natomiast młodzież szkół średnich często już potrafi nazwać różne języki programowania i zdefiniować, gdzie można je zastosować. W związku z ogromną popularnością gier wśród młodszych i starszych uczniów, zaczęło pojawiać się marzenie o tym, żeby tworzyć takie gry samodzielnie. Od marzenia do jego realizacji pozostaje już prosta droga. Trzeba odszukać pozalekcyjne zajęcia z programowania lub poprosić nauczyciela informatyki o stworzenie koła dla pasjonatów programowania lub znaleźć indywidualny kurs online i samodzielnie zdobyć wiedzę i umiejętności. Wadą ostatniego rozwiązania jest brak możliwości konsultowania kolejnych wykonywanych kroków i możliwości przedyskutowania działań z rówieśnikami oraz prowadzącym.

Nasza innowacja pt. **„Progr@muj w zespole”** powstała po to, by zaszcześcić w uczniach pasję do programowania i budować środowisko młodych ludzi, którzy będą mogli się dzielić pomysłami ze swoimi koleżankami i kolegami oraz wiele aktywności realizować we współpracy. Nie jest powszechnie wiadomo, że praca nad projektem IT to praca zespołowa, zaś od różnorodnych talentów członków zespołu i ich efektywnej współpracy zależy dobry produkt końcowy.



W oparciu o tę wiedzę i ogromne zainteresowanie społeczne tematyką programowania (liczne publikacje książkowe, fanpejdże i aktywne grupy w mediach społecznościowych czy blogi IT pokazują skalę zainteresowania tematem) skonstruowaliśmy nasz program, łącząc kurs programowania z doskonaleniem umiejętności komunikacji i pracy uczniów w zespołach.

Naszym podstawowym celem jest stworzenie materiałów informatyczno – społecznych nauczycielom informatyki, którzy chcą uczyć programowania w języku Python zarówno na lekcjach, jak i na różnego rodzaju kołach zainteresowań dedykowanych młodzieży zainteresowanej programowaniem .

Materiał składający się na innowację społeczną pt. **„Progr@muj w zespole”** jest przeznaczony dla:

- nauczycieli informatyki w szkołach średnich uczących w klasach z rozszerzonym programem nauczania z zakresu informatyki,
- nauczycieli i trenerów, którzy uczą programowania na zajęciach pozalekcyjnych lub prowadzą koła IT w szkołach średnich,
- uczniów, którzy chcą samodzielnie lub w zespołach uczyć się języka programowania Python,
- dla rodziców z wykształceniem informatycznym, którzy chcą prowadzić zajęcia z programowania w ramach edukacji domowej lub zajęć pozalekcyjnych dla swoich dzieci i ich przyjaciół,
- dla nauczycieli – wychowawców, którzy mogą wykorzy-



stać materiały projektowe dotyczące komunikacji i współpracy w zespole na lekcjach wychowawczych z młodzieżą,

- dla osób w różnym wieku zainteresowanych programowaniem w ramach edukacji zdalnej, którzy chcą się uczyć samodzielnie lub pracować w zespołach przez siebie stworzonych,
- dla wszystkich pasjonatów, którzy chcą opanować i doskonalić umiejętność programowania w Python i stworzyć pierwszą własną aplikację w tym języku.

Zapraszamy do korzystania z naszych zasobów bez granic (lub ograniczeń) i rozwijania swoich umiejętności programistycznych.



PODZIĘKOWANIA

Ta innowacja nie powstałaby, gdyby nie otwartość i odwaga Hanny Babikowskiej, dyrektor Liceum Ogólnokształcącego im. Adama Mickiewicza w Piastowie.

Dziękujemy za otwarty nad naszym przedsięwzięciem parasol i włączenie materiałów jako zadania szkoły do stosowania w praktyce szkolnej.

Dziękujemy prof. Markowi Piotrowskiemu za opinię, którą wyraził o przygotowanych przez nas materiałach, oraz za wstęp do niniejszego opracowania.

Dziękujemy wszystkim dyrektorom, nauczycielom i uczniom, którzy na poszczególnych etapach tworzenia i testowania innowacji dali nam swój czas i energię, przekazując swoje spostrzeżenia i uwagi: Tomaszowi Niziołowi oraz uczniom II i III klasy LO w Piastowie, Sylwii Piotrowskiej i uczniom klasy I (technik informatyk) w Zespole Szkół Ponadpodstawowych w Białej Rawskiej, Robertowi Celuchowi i uczniom II klasy LO im. M. Kopernika w Iłży oraz III klasy w Zespole Szkół Ogólnokształcących i Technicznych im. Jana Pawła II w Lipsku, Pawłowi Mierzejewskiemu i Michałowi Kurkowi oraz uczniom Katolickiego LO im. bł. B. Markiewicza w Markach, Mykole Pecheniukowi i uczniom III klasy w Niepublicznym LO TeamWork School w Józefowie, Bożenie Rosłoń i Elżbiecie Malinowskiej z Zespołu Szkół nr 1 w Piasecznie, Karolowi Przybyśzewskiemu z Technikum Programistycznego INFOTECH w Białymstoku, Andrzejowi Wyrozembskiemu i Katarzynie Żuchowicz z I LO im. B. Limanowskiego w Warszawie.



Mamy nadzieję, że próbne zajęcia zaowocują wykorzystaniem naszej innowacji podczas lekcji z kolejnymi rocznikami uczniów oraz zainspirują Was do promowania idei współpracy w zespole w Waszych środowiskach edukacyjnych.

Podziękowania kierujemy również do tych nauczycieli informatyki, zaangażowanych w przedsięwzięcia "Lekcja: Enter" oraz "Centrum Mistrzostwa Informatycznego", którzy na wczesnym etapie prac upewnili nas co do słuszności obranego przez nas kierunku.

Dziękujemy całemu zespołowi SENSE Consulting za możliwość, jaką otrzymaliśmy, aby przygotować i sprawdzić w praktyce szkolnej pomysły, które krążyły w naszych umysłach, oraz za cenne lekcje na temat tworzenia i dojrzewania innowacji.

Wreszcie – dziękujemy naszym rodzinom, które z cierpliwością i wyrozumiałością znosiły nasze liczne wieczorne narady i dyskusje.



WSTĘP

Od wielu lat trwają dyskusje na temat kompetencji kluczowych związanych z edukacją przez całe życie. W codziennej pracy, niemalże na każdym stanowisku, spotkać można osoby, które przygotowane są do ciągłego kształcenia i te, które bazują na tym, czego nauczyły się już dawno, czasami bardzo dawno. Wyraźnie zatem widać, że nauczając i ucząc się możemy odnosić sukcesy, zdobywając nowe kompetencje albo doznawać porażek, nie mogąc nadążyć za zmieniającym się światem i jego możliwościami.

W usługach cyfrowych, m.in. tworzeniu nowych urządzeń i oprogramowania, skuteczność odnajdywania optymalnych rozwiązań bardzo często uzależniona jest od umiejętności współpracy niemal wszystkich pracowników. By kształcić tę umiejętność, Fundacja Edukacja na Nowo przygotowała program p.t.: **Progr@muji w zespole**. Uczestnicząca w programie młodzież powinna nie tylko zdobyć umiejętności informatyczne, ale również rozpoznać swoje predyspozycje i rozwijać kompetencje związane ze współpracą. Uczyć się, w jaki sposób budować relacje tak, by zespół mógł odnieść sukces, nieustannie zyskując nowe doświadczenia i nową wiedzę. Nie tylko wykonać bieżące zadanie, ale być gotowym do podejmowania nowych wyzwań, zatem odnieść sukces w dalszej edukacji.

Na marginesie – spór o takie kształcenie, które zapewnia dobrą przyszłość trwa od dawna.

Najczęściej, całkowicie mylnie, łączymy problem uczenia się przez całe życie z XXI wiekiem. W działalności informatycznej przełom nastąpił na początku lat 80. XX wieku, gdy nastąpiła



nowa era wraz z systemem BIOS komputera osobistego IBM. W rzeczywistości problem uczenia się przez całe życie pojawił się po I wojnie światowej wraz z koncepcją progresywizmu edukacyjnego, którego początek umiejscowić można w Stanach Zjednoczonych. Wtedy to – jak nigdy wcześniej – wzrost przemysłu, usług i zasobności obywateli oraz państwa wiązał się z ciągłym pozyskiwaniem wiedzy. W 1919 roku w Waszyngtonie powstało Stowarzyszenie Progresywistów, którego celem była reforma metod uczenia się i nauczania w szkołach państwowych. Sformułowane wtedy zasady i metody pozostają aktualne do dziś. W Polsce zmiana w pojmowaniu edukacji, jako procesu ciągłego, następowała wraz z rozwojem przemysłu w latach 30. XX wieku. Jej pierwszym rzecznikiem był profesor Bogdan Suchodolski, autor określenia „edukacja ustawiczna”.

Powrót do edukacji przede wszystkim umożliwiającej dalszy rozwój, miał nastąpić w 1999 r. Jednak dotychczas rzadko tak się dzieje, by nauczyciele, przekazując wiedzę jednocześnie uczyli jak się uczyć. Kolejne podstawy programowe były nie tylko absurdalnie wypełnione ogromem treści, ale co gorsza nakładały obowiązek uczenia wszystkich tego samego i na jednym poziomie.

Program „**Progr@muji w zespole**” jest przeciwieństwem powyższej sytuacji. Jest skierowany do wybranej młodzieży z klas z rozszerzonym zakresem informatyki liceów ogólnokształcących oraz techników (klas technik – informatyk lub technik programista).

Zajęcia mają poziom zaawansowany. Już na początku uczniowie powinni znać któryś z programów tekstowych lub Python na poziomie podstawowym.



Program zawiera uporządkowane i wyodrębnione materiały, co ułatwia kształcenie umiejętności gospodarowania czasem w trakcie naprzemiennej sekwencji siedmiu „lekcji” po 45 minut oraz pięciu „zajęć domowych” po 60 minut. Dodatkowo materiały edukacyjne podzielono ze względu na ich formę (filmy lub teksty) i umieszczono w odrębnych zasobach, z podziałem na grupę docelową (użytkowników – nauczycieli oraz odbiorców – młodzież):

- nauczyciele otrzymują pomoc w postaci scenariuszy oraz przemyślanych – wzorcowych rozwiązań problemów. Dodatkowo także testy kompetencji i predyspozycji, które mają ułatwić podział uczniów na zespoły oraz ocenić, w jakim stopniu uczniowie oceniają własne umiejętności współpracy i komunikacji przed i po zajęciach zrealizowanych w oparciu o ww. materiały.
- uczniowie otrzymują dokładnie określone zakresy pracy własnej (również w zespole) wraz z kontekstowymi wprowadzeniami. Przewodniki w nowoczesnej formie wideo-tutoriali oraz poradnik dotyczący pracy zespołowej i komunikacji (również w formie wideo). Materiał przekazany uczniom zawiera również wskazówki dotyczące projektów i wzajemnego nauczania (lekcji odwróconej) – najczęstszych form pracy zawodowej programistów.

Co poza tym wyróżnia tę innowację społeczną?

- Autorefleksja – program jest tak skonstruowany, że umożliwi autorefleksję zarówno nad umiejętnościami programistycznymi (po otrzymaniu od nauczyciela kodu umożliwiającego obejrzenie wzorcowego rozwiązania problemu) jak



i informacji o zdobytych umiejętnościach społecznych (po-
miar na początku i na końcu cyklu zajęć).

- Terminarz – w bieżącej realizacji programu ważne jest dotrzymanie terminów – nauka w odpowiednim tempie, wyznaczanym przez nauczyciela i ramy programu zajęć. W przyszłości program może być wykorzystywany w innych uwarunkowaniach, zwłaszcza w częściowej lub całkowitej izolacji będącej wynikiem stanów epidemicznych lub po prostu zwykłego zachorowania i izolacji od świata realnego. Może stanowić również doskonały materiał do rozwijania kompetencji poza klasą, podczas dodatkowych zajęć pozalekcyjnych np. w formie koła zainteresowań, co dodatkowo podnosi poziom motywacji uczniów do osiągnięcia celu programu oraz zwiększa realizm projektu zespołowego, tworząc warunki do podejmowania współpracy z osobami, które uczniowie znają nieco mniej niż kolegów i koleżanki ze szkolnej ławki.
- Dodatkowe kompetencje językowe – z uwagi na angielskojęzyczną dokumentację wykorzystywaną w nauce programowania oraz niewyczerpane źródła wiedzy dostępne w sieci w postaci globalnych serwisów społeczności programistów, program mobilizuje do rozwijania znajomości języka angielskiego w czytaniu i słuchaniu, co warunkuje skuteczność nauki w tym kursie, jak i dalszy rozwój.

Podsumowując

Od wielu lat w szkolnej edukacji informatycznej występuje poważny problem – z jednej strony istnieje na rynku pracy bardzo duże zapotrzebowanie na informatyków, zwłaszcza o wysokich kompetencjach.



Z drugiej strony, liczba uczniów szkół średnich zdających egzamin maturalny z informatyki jest bardzo znikoma. Zatem curriculum i możliwości szkół średnich nie odpowiadają wymaganiom rynku pracy lub/i zainteresowaniom młodzieży.

Projekt jest próbą przełamania tej dysfunkcji. Uczniowie kończący program mają szansę nie tylko zdobyć zawansowane kompetencje informatyczne, ale również zdobyć doświadczenie pracy i nauki zespołowej.

Łatwiej będą mogli otrzymać pracę i jednocześnie kształcić się na kursach dostępnych w Internecie, prowadzonych przez najsilniejsze centra kształcenia (np. amerykańskie uczelnie MIT, Stanford University, University of California—Berkeley). Tym samym program **Progr@muji w zespole** wydaje się być ważnym etapem w drodze do najlepszych zespołów projektowych w Polsce i zagranicą.

Prof. CHAT dr hab. Marek Piotrowski

Wydział Nauk Społecznych Chrześcijańskiej Akademii Teologicznej



PRZEWODNIK

dla **nauczycieli informatyki** dotyczący
rozwoju kompetencji społecznych u uczniów

WPROWADZENIE

Szanowni Nauczyciele,

Przewodnik ma być wsparciem podczas przygotowania i prowadzenia zajęć w ramach projektu „Progr@muj w zespole”. Dlatego też w wielu miejscach nawiązuje do zaproponowanego scenariusza zajęć i materiałów przygotowanych dla uczniów.

Idea naszego projektu jest w dużej mierze oparta na metodzie ***Uczenia się we współpracy (ang. Cooperative Learning)***, rozwijanej m.in. przez Davida i Rogera Johnsonów¹ oraz Spencera Kagana². W ich publikacjach można znaleźć szczegółowe opisy metody, wyniki badań naukowych i bogatą bibliografię. Metoda jest również opisana w polskim piśmiennictwie³.

W przewodniku przedstawiamy podstawowe założenia i informacje dotyczące uczenia się we współpracy. Nauczycielom preferującym przekaz wizualny proponujemy, jako wprowadzenie, krótki filmik Spencera Kagana:

1 <http://www.co-operation.org>, dostęp 06.01.2022

2 <https://www.kaganonline.com>, dostęp 06.01.2022

3 Stanisław Bobula, Norbert Karaszewski, Jakub Kołodziejczyk,
Katarzyna Salamon-Bobińska Sesja II/9,
Nauczanie kooperatywne (uczenie się we współpracy), SEO

W metodzie **Uczenia się we współpracy (ang. Cooperative Learning)**⁴ ucznia postrzega się jako osobę myślącą, zdolną do tworzenia własnych teorii i weryfikowania ich.

Według Spencera Kagana metoda ta opiera się na czterech podstawowych elementach wspólnej nauki:

- **Pozytywna współzależność:**

zespół ustala zasady, zgodnie z którymi członkowie będą pracowali dla osiągnięcia wspólnego celu; wszyscy członkowie czują się odpowiedzialni za siebie i za zespół.

- **Odpowiedzialność indywidualna:**

działania każdego członka mają bezpośredni wpływ na zespół. Wpływ ten może być pozytywny lub negatywny. Każdy członek zespołu czuje się odpowiedzialny za realizację swojego zadania na rzecz wspólnego celu.

- **Równe uczestnictwo:**

wszyscy członkowie mają równe szanse na uczestnictwo w pracach zespołu. Bardzo ważną częścią tej zasady jest to, że praca powinna być sprawiedliwie dzielona od samego początku.

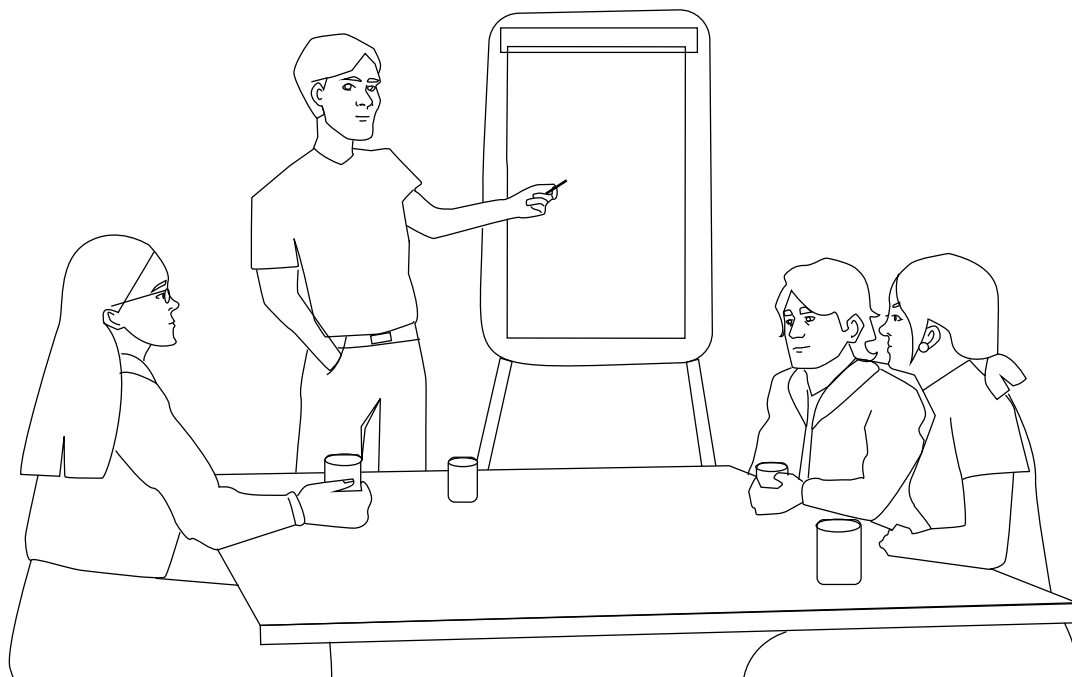
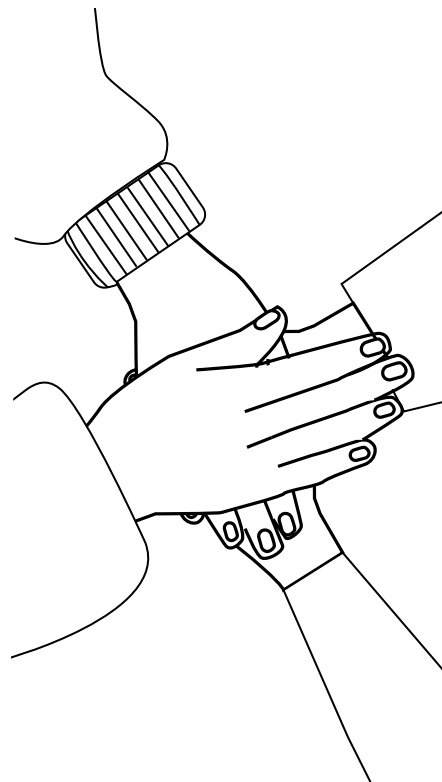
- **Interakcja:**

wszyscy członkowie zespołu muszą się ze sobą komunikować: dzielić się swoimi opiniami, wyrażać emocje dotyczące pracy i wspólnie podejmować decyzje. Jeśli brakuje interakcji, zespół może się rozpaść, co uniemożliwi osiągnięcie celu.



Po zakończeniu wspólnej pracy nad zadaniem projektowym członkowie zespołu dzielą się ze sobą wrażeniami o tym, jak im się współpracowało, co było dobre, a co nie. To bardzo ważny element procesu uczenia się, pozwalający przyswoić, utrwalić i zintegrować nabytą wiedzę oraz umiejętności społeczne.

Podsumowując: uczenie się we współpracy w porównaniu z metodą tradycyjną daje uczniom możliwość bardziej efektywnego przyswajania wiedzy z konkretnej dziedziny, jak i nabywania umiejętności z obszaru rozwoju osobistego takich jak: słuchanie, mówienie, analizowanie, wyciąganie wniosków, szanowanie odmiennych poglądów, szanowanie koleżanek i kolegów, branie odpowiedzialności za siebie i swoją pracę.



Według badań przywołanych we wspomnianej wyżej pracy uczniowie uczący się we współpracy:

- mają większe zdolności rozwiązywania problemów i wykazują rozwinięte strategie kognitywnego⁵ rozumowania,
- lepiej zapamiętują i przekazują skomplikowany materiał,
- wykorzystują wiedzę w praktyce (zadanie szkoły zapisane w podstawie programowej MEIN z 2017r.)
- lepiej posługują się technologią komputerową,
- lepiej rozumieją tekst, pisownię i werbalną ekspresję,
- uzyskują większą samodzielność i jednocześnie rozwijają zachowania kooperatywne,
- wykazują większą tolerancję, więcej zachowań wspomagających i koleżeńskich wobec przedstawicieli innych ras/grup etnicznych lub odmiennej płci,
- z niepełnosprawnościami i wykluczonych społecznie,
- mają wyższe poczucie własnej wartości,
- mają niższy poziom lęku i stresu,
- wykazują większy poziom zadowolenia.

5 kognitywny «mający związek z poznawaniem świata lub badaniem procesów poznawczych», *Słownik języka polskiego PWN*.

Uczenie się we współpracy bardzo dobrze koresponduje z zasadami **PBL (ang. Problem Based Learning)**, czyli uczenia się poprzez rozwiązywanie problemów. Według twórcy PBL Howarda Barrowsa⁶ metodę PBL charakteryzuje sześć elementów:

- koncentracja na odbiorcy – uczniu,
- uczniowie pracują w małych zespołach,
- rozwiązywane problemy/zadania mają wymiar praktyczny i odniesienie do życia codziennego,
- uczniowie, współpracując ściśle z innymi członkami zespołu, rozwijają kompetencje społeczne,
- uczniowie zdobywają wiedzę, narzędzia i informacje zarówno poprzez pracę zespołową jak i indywidualną.
- nauczyciel wspomaga proces komunikacji wewnątrz zespołów – jest facylitatorem.

Jest to metoda skoncentrowana na odbiorcy – czyli uczniu. Proces nauczania według filozofii **PBL** jest ściśle związany z określeniem lub postawieniem problemu, zadaniami, które należy rozwiązać. Wiedza jest ukryta w zadaniu, a cele kształcenia są realizowane podczas prac nad jego zrealizowaniem. Uczniowie w zespołach pracują nad rozwiązaniem praktycznego zagadnienia, znajdującego swoje miejsce w realiach dnia codziennego. Zapotrzebowanie na taką właśnie metodę wybrzmiało dobitnie w wypowiedziach uczniów podczas wywiadów, przeprowadzonych w trakcie przygotowywania niniejszego przewodnika. Zainteresowanych metodą **Problem Based Learning** zachęcam do lektury artykułu⁷ omawiającego metodę, rezultaty jej stosowania oraz podającego bibliografię.

⁶ *Problem-Based Learning: An Approach to Medical Education*, Howard Barrows, 1980

⁷ *Problem-Based Learning*, Jakub Szczepaniak, Iwona Wróblewska, Forum Akademickie nr 07-08/2012, <https://prenumeruj.forumakademickie.pl/fa/2012/07-08/problem-based-learning/>, dostęp 09.01.2022

Szanowni Nauczyciele, aby wszystkie te elementy faktycznie zadziałały i dały trwałe efekty, Państwa rola staje się nieco odmienna od tej w tradycyjnym modelu nauczania. Zainteresowanych pogłębieniem tematu zachęcam do lektury artykułu i książki Spencera Kagana. Poniżej przedstawiam cztery role nauczycieli:

■ **Planista** – uczenie się we współpracy wymaga od nauczyciela wizji i planu tego, co będzie podstawą działań na lekcji, a także zakresu samodzielnej pracy ucznia poza czasem lekcyjnym. Do nauczyciela należy także zaplanowanie i zabezpieczenie technicznej strony zajęć.

■ **Tutor/facilitator** – podczas zajęć wprowadzających (patrz: lekcja 0 w scenariuszu zajęć) dzieli uczniów na zespoły, omawia zasady pracy i oceniania, określa czas, w którym należy zakończyć pracę itp. Przedstawia także swoją rolę tutora/facilitatora – to ważne dla klarowności sytuacji. Zachęca też uczniów do spisania kontraktu regulującego relację nauczyciel – uczniowie, a także kontraktów w zespołach, takich, które ułatwią im działanie. Zadania i role w zespole uczniowie przydzielają sobie sami, bo to ważna część procesu uczenia się, ale może się zdarzyć, że wsparcie i rada nauczyciela będą niezbędne, gdy uczniowie o to poproszą. Warto też upewnić się, że zadania do wykonania rozdzielone zostały równomiernie wśród członków zespołu.

Podczas lekcji nauczyciel w roli tutora obserwuje interakcje uczniów – dzięki temu wie, które zespoły potrzebują jego wsparcia.

■ **Arbiter** – rozstrzygający spory w sytuacjach konfliktowych. One będą się zdarzać, jak zwykle w życiu. Rozwiązywanie sytuacji konfliktowych to doskonała okazja do pokazania skuteczności otwartej komunikacji i szacunku dla stron, empatii i przy okazji, do ugruntowania autorytetu nauczyciela.

■ **Ewaluator** – po zakończeniu pracy nauczyciel oceni ją w dwóch wymiarach:

- **technicznym** (osiągnięcie założonego celu, poprawność funkcjonalna rozwiązania, pozytywny code review etc.),
- **społecznym** (samoorganizacja zespołu, wykonywanie zadań w ramach przypisanych ról, branie odpowiedzialności za siebie i zespół, umiejętności komunikacji i rozwiązywania konfliktów etc.).

Przed rozpoczęciem pracy zespołów i po jej zakończeniu nauczyciel użyje przygotowanego testu kompetencji społecznych (patrz: następny rozdział) i na tej podstawie będzie mógł ocenić efektywność całego procesu.

Warto tu przypomnieć, że nie ma lepszego sposobu wsparcia dla utrwalenia wiedzy i umiejętności, niż dobre słowo i pochwała w odpowiednim momencie. Zakończenie pracy to także wyśmienita okazja do stworzenia przestrzeni dla zespołów, aby mogły skomentować projekty koleżanek i kolegów, a także dla wszystkich uczniów, aby mogli ocenić zajęcia, sposób ich prowadzenia, co im pomagało, co utrudniało pracę, itd. To także doskonała okazja dla nauczyciela, żeby dostać informację zwrotną od swoich uczniów.

ZESPOŁY

Skład zespołów ustali nauczyciel w oparciu o wyniki krótkiego testu oraz własnej znajomości uczniów. Uczniowie nie mają wpływu na ten proces. Myślą przewodnią jest, aby uczniowie nauczyli się pracować w zespołach, na których dobór nie mają wpływu. Taka sytuacja będzie im często towarzyszyć w pracy, którą podejmą jako dorośli. Jest to więc dla nich doskonały poligon, możliwość przećwiczenia w bezpiecznych warunkach szkolnych, umiejętności współpracy z osobami, które niekoniecznie dają największą sympatię.

S. Kagan w swojej książce⁴ i w licznych artykułach² przedstawia optymalny, według niego, skład zespołu. Zespoły powinny być heterogeniczne (mieszane) pod względem poziomu umiejętności, płci i typów osobowości, czyli jak najbardziej różnorodne. Za takim podejściem przemawia kilka argumentów:

- jeśli w każdym zespole jest osoba o wysokich umiejętnościach, prawdopodobnie będzie ona w stanie wyjaśnić pozostałym ewentualne niejasności we wskazówkach nauczyciela lub w pomocniczym materiale wideo i zapewnić wykonanie zadania. Znakomicie poprawia to płynność zarządzania klasą podczas zajęć;
- w przypadku zespołów jednorodnych pod względem poziomu umiejętności istnieje niebezpieczeństwo powstania zespołów zwycięzców i przegranych, i tym samym powstania problemów klasowych związanych ze statusem i wzajemnym szacunkiem;

- im więcej różnorodności, tym więcej uczniowie mogą się od siebie nauczyć;
- tzw. korepetycje rówieśnicze to często najskuteczniejszy sposób nauki dla mniej zdolnych członków zespołu. Dodatkowym zyskiem jest to, że „dający korepetycje” często zyskuje przynajmniej tyle samo, co biorący je;
- zespoły heterogeniczne poprawiają umiejętności społeczne u wszystkich, zwłaszcza u osób osiągających dobre wyniki. Zwykle nie mają one problemu z bardziej skomplikowanymi treściami. Ich największym obszarem rozwoju są często relacje międzyludzkie – gdy uczniowie pracują w mieszanych zespołach, uczą się przekazywania wiedzy, cierpliwości, zachęcania i wzajemnego szacunku;
- optymalna liczba osób w zespole według Kagana to cztery, bo czteroosobowe zespoły maksymalizują i wyrównują aktywne uczestnictwo w porównaniu z każdą inną liczebnością.

Zainteresowanych odsyłam do artykułu⁸. Również Dr Meredith Belbin, autor teorii ról w zespole, uważa, że idealna wielkość zespołu to cztery osoby⁹ – oczywiście, jeśli to możliwe, w parycie płci, czyli dwie dziewczyny i dwóch chłopców. Jeśli nie ma takiej możliwości, zespół powinien być jednorodny pod względem płci. I jeszcze raz oczywiste jest, że czasem nie ma innej możliwości i zespół będzie liczył np. 3 osoby.

■ NO TO DZIELIMY...

Pierwszym elementem w tym procesie będzie wynik krótkiego testu predyspozycji. Test został opracowany przez Wiktora Wołoszkę i z powodzeniem wykorzystany w jednej z jego gier szkoleniowych¹⁰.

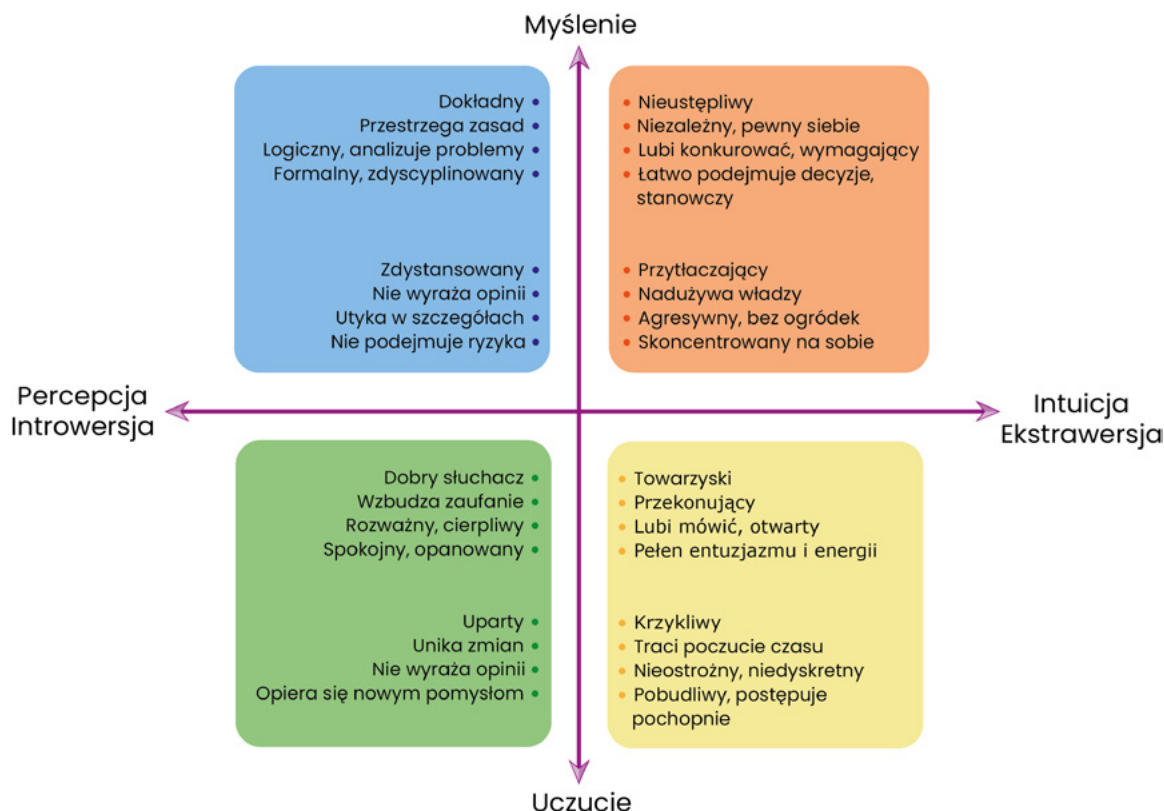
Test bazuje na modelu stworzonym przez Carla Gustava Junga, jednego z twórców podwalin dla współczesnej psychologii, rozwiniętym i udoskonalonym przez wielu badaczy, zgodnie z którym ludzie posiadają odmienne predyspozycje, przez co postrzegają te same sytuacje z różnych perspektyw.

Przyjęcie do wiadomości, zrozumienie tej odmienności, jest kluczowe dla efektywnego komunikowania się i współpracy w zespole. Żeby efektywnie porozumiewać się i współpracować z innymi ludźmi, trzeba najpierw spróbować spojrzeć na świat z ich perspektywy.

8 https://www.kaganonline.com/free_articles/dr_spencer_kagan/295/Teams-of-Four-Are—Magic!

9 <https://www.belbin.pl/role-zespolowe-belbina/>, dostęp 11.01.2022

10 <https://graszkoleniowa.pl>, Wiktor Wołoszko



Współcześnie obowiązujący model obejmuje cztery obszary, przedstawiające cztery podstawowe predyspozycje ludzkich zachowań. Obszary te to ćwiartki układu utworzonego z dwóch przecinających się prostopadłych osi: Percepcja – Intuicja oraz Myślenie – Uczucie. Przykładową graficzną ilustrację modelu przedstawia rysunek. Obszary przypisane poszczególnym predyspozycjom oznaczono kolorami: czerwonym, żółtym, zielonym i niebieskim. Literatura opisująca szczegółowo model jest szeroko dostępna. Obrazowo i przystępnie, w kontekście efektywnej komunikacji przedstawiono go tutaj¹¹.

W zastosowanym teście powyższymi kolorami oznaczono grupy przymiotników opisujących zachowania korespondujące z predyspozycjami z modelu Junga. Każdy z uczniów wybiera ten zestaw przymiotników, który według niego najtrafniej go opisuje oraz drugi w kolejności pod względem trafności opisu. W optymalnej sytuacji każdy członek zespołu reprezentuje inny typ predyspozycji. W przypadku nierównomiernego rozkładu typów predyspozycji w klasie bierzemy pod uwagę drugi wybór.

11 <https://4grow.pl/taxonomy/term/5170>



Przypominam, że podziału na zespoły dokonuje nauczyciel na podstawie powyższego testu oraz znajomości uczniów, ich poziomu umiejętności i wiedzy, z uwzględnieniem parytetu płci. Należy więc zadbać, aby w jednym zespole, tak jak zaleca Kagan, nie znalazły się same osoby najbardziej lub najmniej zaawansowane, jeśli chodzi o wiedzę i umiejętności dotyczące programowania. Będzie więc potrzebny czas do zastanowienia. Na pewno pomoże w tym przygotowana (np. w: Libre Office Calc, MS Excel, Google Sheet) lista uczniów zawierająca wynik testu (1 i 2 wybór), ocenę informatycznego zaawansowania, płeć i ewentualnie inne czynniki, które mogą być istotne (np. ryzykowne jest, aby w jednym zespole znalazły się osoby silnie dominujące lub jawnie ze sobą skonfliktowane).

Wyłoniony przez nauczyciela zespół będzie potrzebował zapewne trochę czasu, aby się zintegrować. Należy zachęcić uczniów do kontaktu, spotkania w czasie pozalekcyjnym, a także zaproponować, aby zespoły nadały sobie nazwy, np. w nawiązaniu do tematu swojej pracy – to ważny moment integrujący, dający poczucie identyfikacji i zaspokajający potrzebę przynależności.

ZASADY PRACY – KONTRAKT

Sformułowanie i przekazanie uczniom zasad, według których będzie przebiegała praca w ramach projektu, to ważna część procesu budowania ich świadomej odpowiedzialności, poczucia podmiotowości oraz nowego wymiaru relacji między nauczycielem a nimi, opartej na wzajemnym szacunku. Część tych zasad ma charakter bardziej formalny, wynikający z przyjętej formuły zajęć i treść tej części kontraktu proponuje nauczyciel (znajdzie je w dalszej części tekstu), natomiast bardzo ważne jest, aby wszystkie inne zasady zostały ustanowione wspólnie, a uczniowie mogli wnieść swoje zapisy. Wtedy każda ze stron może się do takiego kontraktu odwołać dzięki elementarnej identyfikacji z jego treścią, jeśli zaistnieje wymagająca tego sytuacja. Podczas dyskusji nauczyciel powinien starać się spojrzeć na problemy również z perspektywy ucznia.

Poniżej kilka sugestii dla nauczycieli dotyczących samego procesu:

- pokaż uczniom sens kontraktu – spisujemy go po to, aby nasza praca podczas zajęć była efektywna, przebiegała możliwie gładko, a cel każdego zespołu mógł być osiągnięty;
- uświadom uczniom, że obie umawiające się strony mają wpływ na zasady pracy. Zapytaj po prostu co, ich zdaniem, powinno zostać zapisane. Przedyskutujcie to i podejmijcie stosowne decyzje;
- zaproponuj kategorię zasad, które znajdują się w kontrakcie, np.:
 - co robimy – szanujemy się, angażujemy, wykonujemy swoje zadania itp.;
 - czego nie robimy – nie spóźniamy się, nie przeszkadzamy innym zespołom itp.

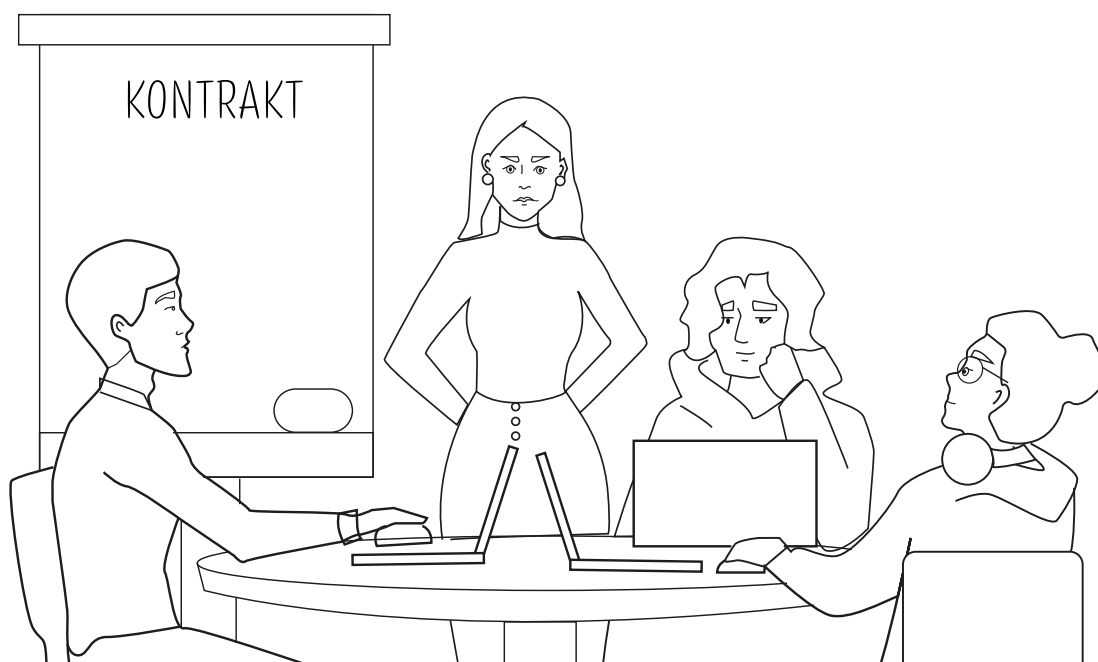
Nauczyciel może też skorzystać z sugestii zawartych w książce Ewy Góralczyk¹².

Kontrakt warto podpisać, bo swoim podpisem potwierdzam: **tak, zgadzam się na to, co jest zapisane**. Podpisany kontrakt dobrze jest umieścić w sali, w której odbywają się zajęcia, po to, żeby zawsze można było do niego sięgnąć.

Poniżej formalne zasady obowiązujące podczas zajęć „Progr@muj w zespole”:

- Uczennice/uczniowie pracują w zespołach. Skład zespołów ustala nauczyciel na podstawie testu i swojej znajomości uczniów.
- Każdy zespół samodzielnie wybiera temat swojego zadania i jest odpowiedzialny za jego wykonanie.
- Przed przystąpieniem do pracy uczennice/uczniowie wypełniają test umiejętności współpracy i komunikacji w zespole.
- Uczennice/uczniowie otrzymują materiały wspierające: przewodnik dla uczniów, materiały wideo.
- Czas trwania zajęć to 13 godzin lekcyjnych podczas zajęć szkolnych i praca własna w czasie pozalekcyjnym.
- Uczennice/uczniowie w każdym zespole wspólnie dokonują podziału pracy koniecznej do wykonania zadania. Każdy członek zespołu jest zaangażowany i ma do wykonania swoją część zadania.
- Zespoły pracują samodzielnie.
- Nauczyciel służy radą i wsparciem, jeśli zostanie o nie poproszony.
- Uczennice/uczniowie mają prawo do własnych opinii dotyczących zadania i ponoszą pełną odpowiedzialność za skutki podjętych przez zespół decyzji.
- Po zakończeniu pracy nad zadaniem uczennice/uczniowie wykonują ponownie test umiejętności współpracy i komunikacji w zespole.
- Wykonana praca podlega ocenie przez nauczyciela. Ocenie podlega osiągnięty rezultat programistyczny oraz praca uczniów w zespole.

Kontrakt może również zabezpieczyć nauczyciela przed ewentualnym przeniesieniem zasad pracy z projektu na pozostałe lekcje prowadzone w sposób tradycyjny, jeżeli taka będzie jego intencja. Może on wtedy wpisać klauzulę, że zapisy dotyczą tylko i wyłącznie pracy podczas zajęć „Progr@muj w zespole”.



W tym miejscu raz jeszcze przypominamy, aby nauczyciel zachęcił uczniów do zawarcia osobnych kontraktów w zespołach. To istotny element w procesie przyjmowania odpowiedzialności zarówno za siebie jak i za zespół, a jednocześnie ugruntowanie poczucia przynależności i identyfikacji z zespołem. Te kontrakty uczniowie powinni sformułować samodzielnie, aczkolwiek rada nauczyciela, jeśli o nią poproszą, może być cenna.

Przykład takiego kontraktu znajdziesz poniżej.

KONTRAKT

- ROBIMY WSZYSTKO, ABY NASZ CEL ZOSTAŁ OSIĄGNIĘTY,
- DOTRZYMUJEMY SŁOWA, WYKONUJEMY NA CZAS SWOJE ZADANIA,
- REALIZUJEMY PODJĘTE DECYZJE,
- PRZYSZCZYM PUNKTUALNIE NA SPOTKANIA,
- KOMUNIKUJEMY SIĘ WEDŁUG USTALONYCH ZASAD,
- SYGNALIZUJEMY ZESPOŁOWI PROBLEMY Z WYKONANIEM SWOJEGO ZADANIA, W RAZIE POTRZEBY PROSIMY O POMOC,
- W RAZIE POTRZEBY POMAGAMY SOBIE, ALE NIE WYRĘCZAMY W PRACY,
- W PRZYPADKU ODMIENNYCH OPINII ZNAJDUJEMY KONSEN-SUS,
- KAŻDEGO SŁUCHAMY Z TAKĄ SAMĄ UWAGĄ, NIE PRZERYWAMY MÓWIĄCEMU,
- SZANUJEMY SIĘ, NIE OBRAŻAMY NIKOGO, NIE ROBIMY FOCHÓW,
- KONFLIKTY STARAMY SIĘ ROZWIĄZYWAĆ SAMI, W RAZIE POTRZEBY PROSIMY NAUCZYCIELA O POMOC.

Spisany kontrakt powinien zostać podpisany przez wszystkich członków zespołu.

ROLE W ZESPOLE

Zwykle, gdy pojawia się temat podziału i objęcia ról w zespole, pojawia się również nazwisko wspomnianego już Mereditha Belbina. Opracował on teorię ról w zespole i dowiódł, że nie tylko wiedza i umiejętności, ale również typ osobowości, preferencje i styl komunikacji mają wpływ na efektywność zespołów¹³. Nie jest naszą intencją przytaczanie tutaj teorii Belbina i jedynie wspomnimy, że na podstawie badań i obserwacji pracy różnych zespołów opisał on trzy profile zachowań ich członków. Są to:

- **role zadaniowe** – osoby, które w pracy w zespole mają silne ukierunkowanie na realizację powierzonych zadań;
- **role socjalne, zorientowane na ludzi** – osoby, które najlepiej funkcjonują i działają, gdy mają obok innych członków zespołu;
- **role intelektualne** – osoby, których zachowanie jest ukierunkowane na myślenie, rozważanie oraz analizowanie.

W ramach tych trzech profili Belbin wyróżnił i nazwał 9 ról zespołowych. Są one według Belbina stałe i oznaczają skłonność do określonych zachowań, nawiązywania współpracy i kontaktów oraz komunikowania się we właściwy danej osobie sposób. Jedna osoba zwykle może przyjąć dwie, trzy role.

Predyspozycje do przyjmowania tych ról można zbadać przy pomocy odpowiedniego, dość rozbudowanego testu. W sytuacji, gdy mamy potrzebę, możliwość i swobodę kompletowania zespołu, informacja o predyspozycjach potencjalnych kandydatów pozwala na taki ich wybór, aby wszystkie role zostały obsadzone.

W pomysłach na zajęcia, w sposób zamierzony odtwarzamy sytuację, jaką często można spotkać w realiach mniejszych organizacji, gdy zespół ma do wykonania określone zadanie, a możliwość swobodnego doboru jego członków jest ograniczona, przypadkowa lub wcale jej nie ma. Wtedy role tylko częściowo mogą być zgodne z preferencjami, a pewna ich część jest wymuszona sytuacją.

13 M. Belbin, *Twoja rola w zespole (Team Roles at Work)*, Gdańskie Wydawnictwa Psychologiczne, Gdańsk 2008

Dlatego tak ważna jest różnorodność predyspozycji wśród członków, o której wspomnieliśmy przy okazji podziału na zespoły. Zwiększa ona po prostu szanse na objęcie większości zidentyfikowanych ról i w rezultacie, na efektywną pracę zespołu.

Role zostaną objęte przez uczniów samodzielnie podczas pierwszego spotkania zespołu. Nauczyciel może zostać poproszony o radę, ale zachęcamy do powściągliwości, aby nie zwalniać zainteresowanych z odpowiedzialności za podejmowane decyzje.

Uczniowie muszą samodzielnie rozdzielić między siebie zadania do wykonania. Część z tych zadań będzie bardziej zgodna z ich naturalnymi preferencjami i predyspozycjami, a część mniej, tak jak to zwykle bywa w życiu.

Rolą nauczyciela jest monitorowanie, aby każdy członek zespołu miał przydzielone zadania i był za nie odpowiedzialny.

KONFLIKT

Konflikty mogą pojawić się w każdym momencie i w każdej zbiorowości ludzkiej. Są po prostu naturalnym procesem społecznym zachodzącym między ludźmi.

Najczęściej mają swoje źródła w różnych wyznawanych wartościach, sprzecznych poglądach, celach lub interesach.



Konflikty w zespole mogą mieć charakter konstruktywny lub destrukcyjny.

Charakter konstruktywny

mają np. spory o możliwe rozwiązania problemu czy zadania, prezentacja sprzecznych opinii i poglądów. I jeśli są one traktowane jako zbiór dostępnych rozwiązań, prowadzą w efekcie do stworzenia nowych jakości i mogą być pozytywnym bodźcem w procesie osiągnięcia celu zespołowego.

Konstruktywne rozwiązywanie konfliktów prowadzi do optymalnych rozwiązań typu wygrany – wygrany. W tym procesie jest przestrzeń na zastosowanie wszelkich dostępnych metod: negocjacji, mediacji, facylitacji i arbitrażu.

Osiągnięcie porozumienia zawsze jest najlepszym rozwiązaniem. Czasami jest ono poprzedzone długimi negocjacjami pochłaniającymi czas i energię zespołu, ale za to przynosi znaczące korzyści w przyszłości. Jeżeli wszyscy członkowie zespołu wypracowywali decyzję, jest ona przez wszystkich akceptowana i może zostać sprawnie wprowadzona w życie. Warto ten mechanizm pokazać uczniom i zachęcić, aby w przypadku zaistnienia spornych kwestii wszyscy członkowie zespołu angażowali się w osiągnięcie konsensusu.

Poniżej kilka wskazówek, które im w tym pomogą:

- Słuchaj uważnie i dopytuj o przyczyny odmiennego stanowiska, co naprawdę za nim stoi. Bądź otwarty na wszystkie pojawiające się idee.
- Wyjaśnij na początku dyskusji, dlaczego jako zespół musicie teraz podjąć decyzję i czego ona będzie dotyczyć.
- Zachęcaj wszystkich członków zespołu do aktywnego udziału w dyskusji. Nie zakładaj, że milczenie oznacza zgodę.
- Postrzegaj różnorodność opinii i stanowisk jako rzecz naturalną i pozytywną, zwiększającą pulę danych, którą zespół może użyć w procesie podejmowania decyzji.
- Nie ulegaj pokusie rezygnacji ze swojego sposobu myślenia tylko po to, żeby uniknąć konfliktu.



- Nie ulegaj pokusie wyszukiwania jedynie takich argumentów, które potwierdzą twój punkt widzenia. Szukaj takich, które będą łącznikiem do pomysłów innych członków zespołu.
- Jeśli masz większą wiedzę na temat niż pozostali członkowie zespołu, zachowaj powściągliwość i poczekaj do momentu, aż oni się wypowiedzą.
- Zadbaj, abyście mieli wystarczająco dużo czasu na dyskusję i podjęcie przemyślanej decyzji.
- Sprawdź i poproś o to pozostałych członków, czy każdy rozumie, na czym polega wasza decyzja, dlaczego jest najlepsza.

Dużo trudniejsze do opanowania są konflikty destrukcyjne, które zwykle są spowodowane wrogim działaniem, złością, poczuciem skrzywdzenia, nieuszanowania konkretnego członka zespołu. Czasami też bywają one przeniesione z obszaru towarzyskiego. O tego rodzaju konflikcie warto chwilę porozmawiać przed rozpoczęciem pracy i uświadomić uczniom, że się może pojawić oraz, że zlekceważenie go lub ukrycie redukuje szansę na efektywną pracę zespołu. Sposób postępowania w przypadku jego zaistnienia można zapisać w kontrakcie i warto to zasugerować uczniom.

A poniżej garść wskazówek, prosto ze szkoły¹⁴, które być może, przydadzą się nauczycielowi w przypadku mediacji między stronami takiego destrukcyjnego konfliktu. A więc Nauczycielu:



- słuchaj uważnie dwóch stron, bezstronnie, nie zajmuj stanowiska,
- słuchaj aktywnie, utrzymuj kontakt wzrokowy ze stronami konfliktu w zależności od tego, która strona mówi, parafrazuj, dopytuj, odzwierciedlaj uczucia,
- dowartościowuj, np. **to, co mówisz, jest ważne...**,
- porządkuj, np. **do tej pory ustaliliśmy, że...**,
- zadaj pytanie: **jak chcielibyście zakończyć konflikt?**,
- pytaj, jakie są oczekiwania wobec strony przeciwnej,
- zachowuj neutralność, nie narzucaj rozwiązania, akceptuj przyjęte rozwiązania i pomagaj zastanowić się, czy przyjęte rozwiązania są wykonalne,
- pytaj obie strony konfliktu, co sądzą o sposobie jego zakończenia,
- jeżeli strony chcą się przeprosić, pytaj, jak rozumieją słowo przepraszam, bo ważne, żeby przeproszenie było autentyczne,



- obserwuj strony konfliktu; jeżeli widzisz, że między uczniami jest napięcie, dopytaj bez zniecierpliwienia, **co jeszcze jest do wyjaśnienia?**
- jeżeli nie masz czasu, spytaj: **czy możemy z tym poczekać np. do jutra?**
- jeżeli masz czas i emocje opadły, spytaj uczniów, jak inaczej mogli się zachować, żeby do konfliktu nie doszło.

SŁÓW KILKA O KOMUNIKOWANIU SIĘ

Podstawową, niezbędną umiejętnością, umożliwiającą rozwijanie innych kompetencji społecznych jest umiejętność komunikowania się czyli tzw. efektywna komunikacja interpersonalna.

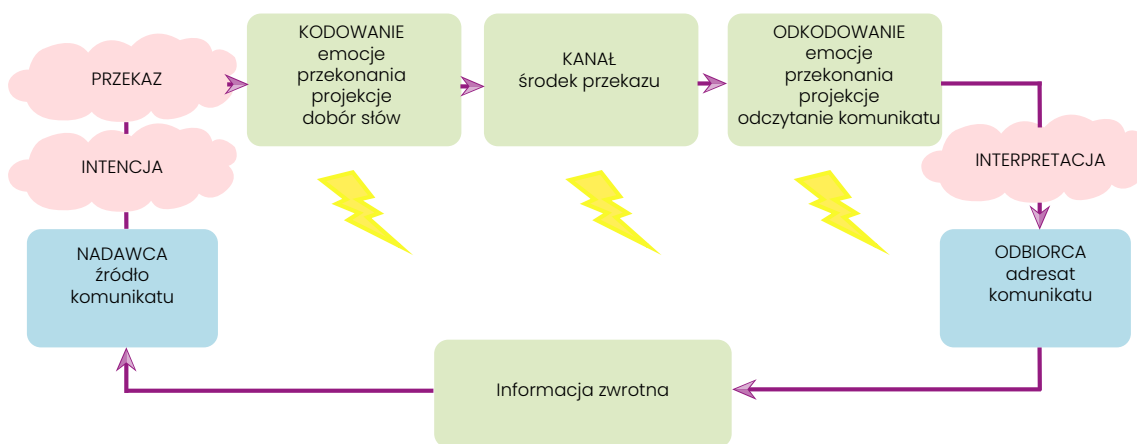
Komunikowanie się to psychologiczny proces, w którym przekazujemy i otrzymujemy informacje, kontaktując się z drugą osobą. Tak naprawdę wszystko w tym kontakcie jest komunikacją. Nasze myśli, opinie czy relacje przekazujemy słowami, tonem głosu, mimiką, postawą ciała, ubiorem i nie tylko. Samo mówienie nie wystarczy, aby się skutecznie komunikować. Komunikując się z drugim człowiekiem, możesz mówić, słuchać i pytać. Zachęcamy, aby nauczyciel na własny użytek przeprowadził autodiagnozę, jaki procent w jego codziennej komunikacji ma mówienie, słuchanie i zadawanie pytań. To oczywiste, że będąc nauczycielem, dużo mówisz. A jak dużo słuchasz (z ciekawością i otwartością na rozmówcę)? Jak wiele pytań zadajesz rozmówcy po to, żeby go lepiej poznać i zrozumieć to, co on chce przekazać, żeby dostrzec jego punkt widzenia?



Myszę, że taka autorefleksja jest ważna wobec faktu, że umiejętność słuchania i zadawania pytań jest tą, którą chcemy rozwinąć wśród uczniów podczas trwania naszych zajęć. Zwiększenie udziału uważnego słuchania i zadawania pytań może przynieść zaskakujący wzrost efektywności komunikacji.

Komunikacja z drugim człowiekiem zaczyna się od intencji. Intencja jest tym, co nadawca komunikatu chce osiągnąć bardziej lub mniej świadomie. Intencja to zamysł, myśl przewodnia, pragnienie. Niewłaściwe odczytanie intencji z jednej strony i niespełnione oczekiwania z drugiej gwarantuje nam komunikacyjny kłops.

Bycie świadomym swojej intencji jest niezbędne do tego, aby nasze myśli były ukierunkowane na efektywne przekazanie sensu, a słowa wyraziły dokładnie to, o co nam konkretnie chodzi. Niestety, często w tym momencie pojawiają się emocje, nasze myśli wpadają w błędne koło przekonań, projekcji, nadinterpretacji i obaw, że odbiorca nas nie zrozumie, źle o nas pomyśli i w efekcie zniekształcają nasz przekaz. Ten proces nazywamy kodowaniem naszego komunikatu. Po drugiej stronie jest odbiorca ze swoim analogicznym bagażem projekcji, przekonań i nadinterpretacji, emocji, który usiłuje odkodować nasz przekaz. W efekcie komunikat nadany i odebrany to często zupełnie różne treści.



Kodowanie jest najbardziej krytycznym momentem komunikacji. Albo nadawca utrzyma kontakt ze swoją intencją, będzie jej świadomy i powie, o co konkretnie mu chodzi, albo znacząco obniża szansę na porozumienie. Żeby tego kontaktu z intencją nie tracić, możesz świadomie używać w swojej komunikacji takich sformułowań jak:

Moją intencją jest, aby ...
Chcę zadbać o to, aby ...
Wartością dla mnie w tym jest to, że ...
Zależy mi na tym, aby ...

Taka sama odpowiedzialność za efektywność komunikacji jest po stronie odbiorcy – uważne słuchanie i zadawanie pytań znacząco zwiększa szansę na porozumienie. I tu możesz świadomie używać takich sformułowań jak:

Co jest twoją intencją?
O co chcesz zadbać?
Co w tym dla ciebie jest wartościowe?
Na czym ci zależy?

Pierwszym krokiem do skutecznej komunikacji jest chęć porozumienia się po obu stronach. Nie warto zakładać, że rozmówca zrozumiał, ale że chce zrozumieć. Warto pamiętać, że jako nadawca i odbiorca jednocześnie, każdy z nas jest współodpowiedzialny za efekt komunikowania się.

Dlatego tak ważne jest pogłębianie samoświadomości i kontakt ze swoimi emocjami.

GDY POJAWIAJĄ SIĘ EMOCJE...

...to znaczy, że dzieją się rzeczy ważne. Jeśli to emocje nauczyciela, są to rzeczy ważne dla niego. Jeśli to emocje innej osoby, ucznia albo całej grupy uczniów, klasy, znaczy, że to są rzeczy ważne dla niej/nich. Tylko od nauczyciela zależy, jak je odczyta, jak nimi zarządzi, czy skorzysta z informacji, które mu przynoszą. Emocje mają to do siebie, że pojawiają się dosłownie w jednej chwili. Radzenie sobie z emocjami nie polega na ich tłumieniu i kontrolowaniu, tylko na ich doświadczaniu i przeżywaniu.

Dlatego tak ważna jest umiejętność wyrażenia, jak się w danym momencie czujesz, nazwania, opisanie swoich emocji.

Okazuje się, że najlepszym sposobem na wyciszenie emocji w trudnej sytuacji jest uznanie ich realności, nazwanie ich:

***tak, jestem niezadowolona,
tak, jestem rozczarowany,
tak, widzę, że się denerwujesz.***

Zaprzeczanie im:

***nie ma sensu tak się denerwować,
nie powinnaś czuć się dotknięta,
nie bądź taki urażony,
nie ma co robić tragedii,***

powoduje tylko aktywację mechanizmów obronnych, prowadzi do frustracji, złości, jest odbierane jako lekceważenie, brak szacunku i osądzanie.

Potwierdzenie, nazwanie emocji ma sens, ponieważ:

- czyjeś uczucie, emocja jest po prostu faktem,
- sygnalizuje coś ważnego kryjącego się pod tą emocją,
- dotarcie do tego czegoś, do rzeczywistego problemu, możliwe jest dopiero wtedy, gdy emocja zostanie uznana i zauważona,
- poszanowanie czyichś emocji oznacza okazanie szacunku osobie te emocje przeżywającej, a to jest podstawą dobrej relacji.

Emocje działają jak kierunkowskaz: mierzymy w kierunku tych przyjemnych, wzmagających chęć życia i działania, natomiast aktywnie unikamy tych nieprzyjemnych, osłabiających, destrukcyjnych. Emocje mają więc wartość motywacyjną; można by powiedzieć, że motywacja to emocja¹⁵.

Emocje są też informacją, czy podążamy za tym, co jest dla nas ważne, za swoimi wartościami, czy działamy wbrew sobie? Czy nasze potrzeby są zaspokojone, czy też nie? Wiele potrzeb jest wspólnych dla wszystkich ludzi. Należą do nich potrzeby: akceptacji, docenienia, wysłuchania i bycia zauważonym.

Myślę, że każdy nauczyciel zgodzi się, że są to potrzeby uczniów, dla których jest tutorem. Aby te potrzeby mogły zostać zauważone i zaspokojone, w relacji nauczyciel – uczeń, musi zostać zbudowane zaufanie.

Zaufanie buduje się w oparciu o kluczowe wartości: szacunek, życzliwość, odpowiedzialność. Zaufanie jest niezbędne dla efektywnego komunikowania się.

Dlatego rola nauczyciela jest po prostu kluczowa.

15 R. Gut, A. Gut, *Pokolenie Y. Zarządzanie sobą na Zielonej Ścieżce*. Wyd. Instytut Flashpoint, Wrocław 2016.

BADANIE UMIEJĘTNOŚCI WSPÓŁPRACY I KOMUNIKACJI W ZESPOLE

WPROWADZENIE

Poniższy test został przygotowany na bazie opracowania Science Museum of Minnesota (SMM)¹. Zamysł testu koncentruje się na fakcie, że kluczową kompetencją, na której opiera się współpraca w zespole, jest skuteczne komunikowanie się.

Wyróżniliśmy pięć obszarów w komunikacji w zespole:

1. dzielenie się pomysłami i informacjami (pytania 1-6)
2. precyzja przekazu (pytania 7-12)
3. asertywność (pytania 13-18)
4. otwartość i uważność w zadawaniu pytań (pytania 19-24)
5. koncentracja i uważne słuchanie (pytania 25-28)

O znaczeniu zespołu i wymienionych tutaj umiejętności nauczyciel może przeczytać więcej w Przewodniku dla nauczycieli po kompetencjach współpracy i komunikacji w zespole oraz źródłach zewnętrznych. Należy pamiętać, że: ucząc przedmiotu, w tym wypadku informatyki, warto stosować pracę zespołową – kompetencje społeczne są cennym zasobem u absolwentów podczas poszukiwania zatrudnienia i są bardzo pożądane przez pracodawców.

STRUKTURA TESTU

Test ma formułę samooceny badanego i mierzy łatwość, komfort oraz prawdopodobieństwo użycia konkretnej umiejętności w określonej sytuacji. Te sytuacje są wkomponowane w scenariusz pracy zespołowej, będący kontekstem dla zadanych pytań.

1 Amy Grack Nelson, PhD, *Youth Teamwork Skills Survey: Manual and Survey*, Science Museum of Minnesota, 2018

Wszystkie pytania wymagają odpowiedzi opartych na sześciostopniowej skali.

Pytania są pogrupowane w ww. 5 obszarów.

ZASTOSOWANIE TESTU

Niniejszy test stanowi źródło informacji nt. stanu początkowego i końcowego indywidualnej percepcji ucznia dotyczącej jego kompetencji współpracy i komunikacji w zespole. Należy pamiętać, aby test przeprowadzić w trakcie lekcji wprowadzającej (lekcja 0) oraz lekcji podsumowującej zajęcia (lekcja 12), aby najpełniej zbadać efekt przeprowadzonych zajęć.

Co istotne, test może być wykorzystany do pomiaru samooceny kompetencji w ramach innych zajęć prowadzonych w szkole, które można zaliczyć do kategorii STEM (ang. science, technology, engineering, mathematics – poza informatyką mogą to być zatem: matematyka, fizyka, chemia czy biologia) – zachęcamy nauczyciela, aby promował jego stosowanie przez innych nauczycieli, którym zależy na: wiedzy nt. subiektywnej oceny uczniów posiadanych już kompetencji lub pomiarze skuteczności prowadzonych przez nich projektów opartych na pracy zespołowej. Co ważne, test poza naszymi zajęciami można przeprowadzać modułowo, tj. jeśli dany nauczyciel chce zmierzyć poziom kompetencji np. w ramach 3 z 5 ww. obszarów (które uznaje za istotne z punktu widzenia jego zajęć), to z łatwością może to zrobić bez uszczerbku dla poprawności wyników – trafność wewnętrzna testu została zweryfikowana przez naukowców z SMM. Należy jednak zawsze pamiętać o pozostawieniu scenariusza pracy zespołowej, który rozpoczyna test, oraz zachowaniu kolejności badanych obszarów.

Narzędzie to może być również stosowane do oceny kompetencji młodzieży biorącej udział w zajęciach pozalekcyjnych z zakresu STEM (kółka zainteresowań, kursy weekendowe, szkoły letnie etc.).

Test przeznaczony jest dla młodzieży w wieku od lat 12 do 18.

ZAŁOŻENIA ORGANIZACYJNE

Czas, który nauczyciel powinien przeznaczyć na wypełnienie testu przez uczniów, to nie więcej niż 15 minut (zakładamy, że przeciętnie będzie to 5 do 10 minut). Zachęcamy nauczycieli do korzystania z narzędzi cyfrowych wspomagających ankietowanie; na potrzeby niniejszych zajęć udo-

stępniamy do pobrania szablon przygotowany w MsForms, który można zduplikować i udostępnić uczniom w ramach posiadanego konta Microsoft (uczniowie nie muszą posiadać takiego konta, formularz testu będzie dostępny w przeglądarce). Dzięki narzędziom online nauczyciel będzie mógł na żywo podejrzeć, czy wszyscy uczniowie wypełnili już test.



TEST UMIEJĘTNOŚCI WSPÓŁPRACY I KOMUNIKACJI W ZESPOLE

Przeczytaj najpierw poniższy scenariusz dotyczący pracy w zespole. Pamiętaj o nim, odpowiadając na pytania w ankiecie.

Scenariusz pracy zespołowej

Wyobraź sobie, że bierzesz udział w pewnym projekcie i właśnie zostałeś przydzielony/a do zespołu, pracującego nad zadaniem. W zespole jest jeszcze troje młodych ludzi. W sumie jest was czworo, dwie dziewczyny i dwóch chłopaków. Wszyscy chodzicie do szkół średnich. Osoby z zespołu spotkałeś dzisiaj po raz pierwszy w życiu.

Zanim zaczęliście wspólną pracę, wszyscy uczestnicy projektu przedstawili się imieniem i nazwiskiem oraz podali po pięć ciekawych informacji na swój temat.

Następnie zespół zapoznaje się ze szczegółami zadania i upewnia się, że wszyscy w zespole rozumieją, co mają wykonać. Następnie członkowie zespołu dzielą się ze sobą informacją, jaką posiadają wiedzę z dziedziny, której dotyczy zadanie. Jako zespół stwierdzacie, że potrzebujecie dowiedzieć się więcej na ten temat.

Jako członkowie zespołu dzielicie się pracą przy wyszukiwaniu informacji online, w książkach i czasopiśmie oraz wśród informacji dostarczonych przez organizatorów projektu. Po pewnym czasie spotykacie się ponownie, aby przekazać sobie zgromadzoną wiedzę. Następnie w zespole ustalacie, co z tej wiedzy może być potrzebne do ukończenia projektu. Decydujecie, jakie prace należy wykonać oraz kto z zespołu będzie je wykonywał, a następnie zabieracie się do pracy zespołowej i indywidualnej. Osiągnięcie celu zespołu zależy od zaangażowania i wkładu wszystkich jego członków, jesteście więc ze sobą w bliskim kontakcie.

Wyobraź sobie, że jesteś członkiem opisanego wyżej zespołu. Poniżej są opisane sytuacje i są zadane pytania. Wyobraź sobie, że znajdujesz się

w tych sytuacjach i masz wykonać określone czynności. Odpowiedz na pytania ankiety otwarcie i szczerze. Nie ma dobrych ani złych odpowiedzi na te pytania i nie jesteś za nie oceniany. Pamiętaj, że zespół to ty i trójka innych młodych ludzi.

Pomyśl o sytuacji, w której dzielisz się wyszukanymi informacjami na temat wspólnego zadania, o których żaden/żadna z Twoich koleżanek/kolegów dotąd nie wspomniał/-a.

1. Jak dobrze lub źle byś sobie z tym radził w swoim zespole?

- 1. ☐ bardzo źle
- 2. ☐ źle
- 3. ☐ raczej źle
- 4. ☐ umiarkowanie dobrze
- 5. ☐ dobrze
- 6. ☐ bardzo dobrze

2. Jak komfortowo lub niekomfortowo czułbyś się, wykonując to w swoim zespole?

- 1. ☐ bardzo niekomfortowo
- 2. ☐ niekomfortowo
- 3. ☐ raczej niekomfortowo
- 4. ☐ umiarkowanie komfortowo
- 5. ☐ komfortowo
- 6. ☐ bardzo komfortowo



3. Jak bardzo prawdopodobne lub mało prawdopodobne byłoby, abyś wykonał to w swoim zespole?

1. ☐ bardzo mało prawdopodobne
2. ☐ mało prawdopodobne
3. ☐ niezbyt prawdopodobne
4. ☐ umiarkowanie prawdopodobne
5. ☐ prawdopodobne
6. ☐ bardzo prawdopodobne

A teraz wyobraź sobie, że dzielisz się swoimi pomysłami z trójką swoich koleżanek i kolegów z zespołu.

Pomyśl o sytuacji, w której wyjaśniasz swój pomysł zespołowi.

4. Jak dobrze lub źle byś sobie z tym radził w swoim zespole?

1. ☐ bardzo źle
2. ☐ źle
3. ☐ raczej źle
4. ☐ umiarkowanie dobrze
5. ☐ dobrze
6. ☐ bardzo dobrze



5. Jak komfortowo lub niekomfortowo czułbyś się, wykonując to w swoim zespole?

1. ☐ bardzo niekomfortowo
2. ☐ niekomfortowo
3. ☐ raczej niekomfortowo
4. ☐ umiarkowanie komfortowo
5. ☐ komfortowo
6. ☐ bardzo komfortowo

6. Jak bardzo prawdopodobne lub mało prawdopodobne byłoby, abyś wykonał to w swoim zespole?

1. ☐ bardzo mało prawdopodobne
2. ☐ mało prawdopodobne
3. ☐ niezbyt prawdopodobne
4. ☐ umiarkowanie prawdopodobne
5. ☐ prawdopodobne
6. ☐ bardzo prawdopodobne



Pomyśl o sytuacji, w której pytasz swoje koleżanki i kolegów w zespole, czy zrozumieli Twój pomysł.

7. Jak dobrze lub źle byś sobie z tym radził w swoim zespole?

1. ☐ bardzo źle
2. ☐ źle
3. ☐ raczej źle
4. ☐ umiarkowanie dobrze
5. ☐ dobrze
6. ☐ bardzo dobrze

8. Jak komfortowo lub niekomfortowo czułbyś się, wykonując to w swoim zespole?

1. ☐ bardzo niekomfortowo
2. ☐ niekomfortowo
3. ☐ raczej niekomfortowo
4. ☐ umiarkowanie komfortowo
5. ☐ komfortowo
6. ☐ bardzo komfortowo

9. Jak bardzo prawdopodobne lub mało prawdopodobne byłoby, abyś wykonał to w swoim zespole?

1. ☐ bardzo mało prawdopodobne
2. ☐ mało prawdopodobne
3. ☐ niezbyt prawdopodobne
4. ☐ umiarkowanie prawdopodobne
5. ☐ prawdopodobne
6. ☐ bardzo prawdopodobne

Pomyśl o sytuacji, w której zachęcasz swoje koleżanki i kolegów z zespołu do zadawania Tobie pytań dotyczących twojego pomysłu, aby się upewnić, że dobrze go zrozumieli.

10. Jak dobrze lub źle byś sobie z tym radził w swoim zespole?

1. ☐ bardzo źle
2. ☐ źle
3. ☐ raczej źle
4. ☐ umiarkowanie dobrze
5. ☐ dobrze
6. ☐ bardzo dobrze



11. Jak komfortowo lub niekomfortowo czułbyś się, wykonując to w swoim zespole?

1. ☐ bardzo niekomfortowo
2. ☐ niekomfortowo
3. ☐ raczej niekomfortowo
4. ☐ umiarkowanie komfortowo
5. ☐ komfortowo
6. ☐ bardzo komfortowo

12. Jak bardzo prawdopodobne lub mało prawdopodobne byłoby, abyś wykonał to w swoim zespole?

1. ☐ bardzo mało prawdopodobne
2. ☐ mało prawdopodobne
3. ☐ niezbyt prawdopodobne
4. ☐ umiarkowanie prawdopodobne
5. ☐ prawdopodobne
6. ☐ bardzo prawdopodobne



Pomyśl o sytuacji, w której dzielisz się pomysłem, sądząc że może on się nie spodobać Twoim koleżankom i kolegom.

13. Jak dobrze lub źle byś sobie z tym radził w swoim zespole?

1. ☐ bardzo źle
2. ☐ źle
3. ☐ raczej źle
4. ☐ umiarkowanie dobrze
5. ☐ dobrze
6. ☐ bardzo dobrze

14. Jak komfortowo lub niekomfortowo czułbyś się, wykonując to w swoim zespole?

1. ☐ bardzo niekomfortowo
2. ☐ niekomfortowo
3. ☐ raczej niekomfortowo
4. ☐ umiarkowanie komfortowo
5. ☐ komfortowo
6. ☐ bardzo komfortowo



15. Jak bardzo prawdopodobne lub mało prawdopodobne byłoby, abyś wykonał to w swoim zespole?

1. ☐ bardzo mało prawdopodobne
2. ☐ mało prawdopodobne
3. ☐ niezbyt prawdopodobne
4. ☐ umiarkowanie prawdopodobne
5. ☐ prawdopodobne
6. ☐ bardzo prawdopodobne

Pomyśl o sytuacji, w której przedstawiasz pomysł, który różni się od pomysłu, o którym właśnie skończyliście dyskutować.

16. Jak dobrze lub źle byś sobie z tym radził w swoim zespole?

1. ☐ bardzo źle
2. ☐ źle
3. ☐ raczej źle
4. ☐ umiarkowanie dobrze
5. ☐ dobrze
6. ☐ bardzo dobrze



17. Jak komfortowo lub niekomfortowo czułbyś się, wykonując to w swoim zespole?

1. ☐ bardzo niekomfortowo
2. ☐ niekomfortowo
3. ☐ raczej niekomfortowo
4. ☐ umiarkowanie komfortowo
5. ☐ komfortowo
6. ☐ bardzo komfortowo

18. Jak bardzo prawdopodobne lub mało prawdopodobne byłoby, abyś wykonał to w swoim zespole?

1. ☐ bardzo mało prawdopodobne
2. ☐ mało prawdopodobne
3. ☐ niezbyt prawdopodobne
4. ☐ umiarkowanie prawdopodobne
5. ☐ prawdopodobne
6. ☐ bardzo prawdopodobne

Teraz wyobraź sobie, że Twoje koleżanki i koledzy dzielą się swoimi pomysłami.

Pomyśl o sytuacji, w której prosisz koleżankę/kolegę o wyjaśnienie ich pomysłu w inny sposób, żebyś lepiej mógł go zrozumieć.

19. Jak dobrze lub źle byś sobie z tym radził w swoim zespole?

1. ☐ bardzo źle
2. ☐ źle
3. ☐ raczej źle
4. ☐ umiarkowanie dobrze
5. ☐ dobrze
6. ☐ bardzo dobrze

20. Jak komfortowo lub niekomfortowo czułbyś się, wykonując to w swoim zespole?

1. ☐ bardzo niekomfortowo
2. ☐ niekomfortowo
3. ☐ raczej niekomfortowo
4. ☐ umiarkowanie komfortowo
5. ☐ komfortowo
6. ☐ bardzo komfortowo



21. Jak bardzo prawdopodobne lub mało prawdopodobne byłoby, abyś wykonał to w swoim zespole?

1. ☐ bardzo mało prawdopodobne
2. ☐ mało prawdopodobne
3. ☐ niezbyt prawdopodobne
4. ☐ umiarkowanie prawdopodobne
5. ☐ prawdopodobne
6. ☐ bardzo prawdopodobne

Pomyśl o sytuacji, w której prosisz koleżankę/kolegę o powtórzenie, bo nie jesteś pewien, czy zrozumiałeś właściwie jej/jego pomysł.

22. Jak dobrze lub źle byś sobie z tym radził w swoim zespole?

1. ☐ bardzo źle
2. ☐ źle
3. ☐ raczej źle
4. ☐ umiarkowanie dobrze
5. ☐ dobrze
6. ☐ bardzo dobrze



23. Jak komfortowo lub niekomfortowo czułbyś się, wykonując to w swoim zespole?

1. ☐ bardzo niekomfortowo
2. ☐ niekomfortowo
3. ☐ raczej niekomfortowo
4. ☐ umiarkowanie komfortowo
5. ☐ komfortowo
6. ☐ bardzo komfortowo

24. Jak bardzo prawdopodobne lub mało prawdopodobne byłoby, abyś wykonał to w swoim zespole?

1. ☐ bardzo mało prawdopodobne
2. ☐ mało prawdopodobne
3. ☐ niezbyt prawdopodobne
4. ☐ umiarkowanie prawdopodobne
5. ☐ prawdopodobne
6. ☐ bardzo prawdopodobne



Kolejne pytania dotyczą sytuacji, które mogłyby się wydarzyć podczas wspólnej pracy waszego zespołu.

Pomyśl o tym, jak łatwo lub jak trudno byłoby Tobie wykonać to, co jest opisane poniżej.

25. Słuchaj uważnie koleżanki/kolegi z zespołu, która/-y dzieli się swoim pomysłem, zamiast koncentrować się na tym, co chcesz powiedzieć zespołowi o Twoim pomysśle.

1. ☐ bardzo trudno
2. ☐ trudno
3. ☐ niezbyt trudno
4. ☐ umiarkowanie łatwo
5. ☐ łatwo
6. ☐ bardzo łatwo

26. Skoncentruj się na rozmowie z Twoim zespołem, nie odpływaj myślami.

1. ☐ bardzo trudno
2. ☐ trudno
3. ☐ niezbyt trudno
4. ☐ umiarkowanie łatwo
5. ☐ łatwo
6. ☐ bardzo łatwo



27. Skoncentruj się na tym, co mówi koleżanka/kolega z zespołu, zamiast myśleć o tym, co za chwilę sam powiesz zespołowi.

1. ☐ bardzo trudno
2. ☐ trudno
3. ☐ niezbyt trudno
4. ☐ umiarkowanie łatwo
5. ☐ łatwo
6. ☐ bardzo łatwo

28. Skoncentruj się na tym, co mówi koleżanka/kolega z zespołu, podczas, gdy sam wolałbyś pracować nad swoją częścią zadania zespołowego.

1. ☐ bardzo trudno
2. ☐ trudno
3. ☐ niezbyt trudno
4. ☐ umiarkowanie łatwo
5. ☐ łatwo
6. ☐ bardzo łatwo

ARKUSZ Z KLUCZEM PUNKTACJI

OGÓLNE WSKAZÓWKI I SUGESTIE DLA NAUCZYCIELA

Dla każdego obszaru, opisanego we wprowadzeniu do niniejszego narzędzia badawczego:

1. numerom przypisanym odpowiedziom przypisz identyczną punktację (np. jeśli "bardzo źle" to odpowiedź nr 1, to do tej odpowiedzi przypisz wynik "1 punkt" etc.)
2. zsumuj punkty dla danego obszaru (6 pytań to 6 wyników do zsumowania etc.), a następnie podziel przez liczbę pytań w danym obszarze, by uzyskać średni wynik dla obszaru.

Następnie zsumuj uzyskane średnie, aby uzyskać wynik końcowy; zachowaj jednak wyniki częściowe, gdyż potencjalnie stanowią one cenną informację zwrotną dla uczniów, jak również stanowią źródło wiedzy nt. obszarów, w których uczniowie dostrzegają własne deficyty (może to być wskazówka dla wzmocnienia pewnych akcentów w dalszej pracy zespołowej – zarówno na lekcjach informatyki, jak i pozostałych przedmiotów).

Możesz za pomocą testu ocenić również skuteczność zajęć dla całej klasy/grupy – uśrednij wówczas wyniki uzyskane przez uczniów, sumując ich wyniki indywidualne (zarówno w poszczególnych obszarach, jak i ogółem), a następnie dzieląc je przez liczbę uczniów uczestniczących w zajęciach. Jeśli uznasz to za istotne, możesz agregować wyniki wg określonych cech uczniów (np. płci lub wieku) – pozwoli Ci to ocenić, w jaki sposób zajęcia dopasowane są do tych podgrup i jakie różnice występują między nimi, jeśli chodzi o samoocenę kompetencji współpracy i komunikacji w zespole.

Uwaga: zachowaj indywidualne wyniki dla siebie. Jeśli uznasz, że informacja o wyniku może być cenną informacją zwrotną dla ucznia, oczywiście możesz przekazać uczniowi jego wynik (po zakończeniu testowania). Korzystaj z uogólnionych wyników dla grupy uczniów (np. klasy), aby lepiej projektować kolejne zajęcia w zespołach.

SPOSÓB OBLICZANIA PUNKTACJI DLA PIĘCIU OBSZARÓW

Pytania związane z obszarem 1 – dzielenie się pomysłami i informacją

Pomyśl o sytuacji, w której dzielisz się wyszukiwanymi informacjami na temat wspólnego zadania, o których żaden/żadna z Twoich koleżanek/kolegów dotąd nie wspomniał/-a.

1. Jak dobrze lub źle byś sobie z tym radził w swoim zespole?	
bardzo źle	1 punkt
źle	2 punkty
raczej źle	3 punkty
umiarkowanie dobrze	4 punkty
dobrze	5 punktów
bardzo dobrze	6 punktów

2. Jak komfortowo lub niekomfortowo czułbyś się, wykonując to w swoim zespole?	
bardzo niekomfortowo	1 punkt
niekomfortowo	2 punkty
niezbyt niekomfortowo	3 punkty
umiarkowanie komfortowo	4 punkty
komfortowo	5 punktów
bardzo komfortowo	6 punktów



3. Jak bardzo prawdopodobne lub mało prawdopodobne byłoby, abyś wykonał to w swoim zespole?	
bardzo mało prawdopodobne	1 punkt
mało prawdopodobne	2 punkty
niezbyt prawdopodobne	3 punkty
umiarkowanie prawdopodobne	4 punkty
prawdopodobne	5 punktów
bardzo prawdopodobne	6 punktów

Pomyśl o sytuacji, w której wyjaśniasz swój pomysł zespołowi.

4. Jak dobrze lub źle byś sobie z tym radził w swoim zespole?	
bardzo źle	1 punkt
źle	2 punkty
raczej źle	3 punkty
umiarkowanie dobrze	4 punkty
dobrze	5 punktów
bardzo dobrze	6 punktów

5. Jak komfortowo lub niekomfortowo czułbyś się, wykonując to w swoim zespole?	
bardzo niekomfortowo	1 punkt
niekomfortowo	2 punkty
niezbyt niekomfortowo	3 punkty
umiarkowanie komfortowo	4 punkty
komfortowo	5 punktów
bardzo komfortowo	6 punktów

6. Jak bardzo prawdopodobne lub mało prawdopodobne byłoby, abyś wykonał to w swoim zespole?	
bardzo mało prawdopodobne	1 punkt
mało prawdopodobne	2 punkty
niezbyt prawdopodobne	3 punkty
umiarkowanie prawdopodobne	4 punkty
prawdopodobne	5 punktów
bardzo prawdopodobne	6 punktów

Zsumuj punkty dla obszaru 1 (dla pytań 1-6) i podziel przez 6.

Pytania związane z obszarem 2 – precyzja przekazu

Pomyśl o sytuacji, w której pytasz swoje koleżanki i kolegów w zespole, czy zrozumieli Twój pomysł.

7. Jak dobrze lub źle byś sobie z tym radził w swoim zespole?	
bardzo źle	1 punkt
źle	2 punkty
raczej źle	3 punkty
umiarkowanie dobrze	4 punkty
dobrze	5 punktów
bardzo dobrze	6 punktów

8. Jak komfortowo lub niekomfortowo czułbyś się, wykonując to w swoim zespole?	
bardzo niekomfortowo	1 punkt
niekomfortowo	2 punkty
niezbyt niekomfortowo	3 punkty
umiarkowanie komfortowo	4 punkty
komfortowo	5 punktów
bardzo komfortowo	6 punktów



9. Jak bardzo prawdopodobne lub mało prawdopodobne byłoby, abyś wykonał to w swoim zespole?	
bardzo mało prawdopodobne	1 punkt
mało prawdopodobne	2 punkty
niezbyt prawdopodobne	3 punkty
umiarkowanie prawdopodobne	4 punkty
prawdopodobne	5 punktów
bardzo prawdopodobne	6 punktów

Pomyśl o sytuacji, w której zachęcasz swoje koleżanki i kolegów z zespołu do zadawania Tobie pytań dotyczących Twojego pomysłu, aby się upewnić, że dobrze go zrozumieli.

10. Jak dobrze lub źle byś sobie z tym radził w swoim zespole?	
bardzo źle	1 punkt
źle	2 punkty
raczej źle	3 punkty
umiarkowanie dobrze	4 punkty
dobrze	5 punktów
bardzo dobrze	6 punktów



11. Jak komfortowo lub niekomfortowo czułbyś się, wykonując to w swoim zespole?	
bardzo niekomfortowo	1 punkt
niekomfortowo	2 punkty
niezbyt niekomfortowo	3 punkty
umiarkowanie komfortowo	4 punkty
komfortowo	5 punktów
bardzo komfortowo	6 punktów

12. Jak bardzo prawdopodobne lub mało prawdopodobne byłoby, abyś wykonał to w swoim zespole?	
bardzo mało prawdopodobne	1 punkt
mało prawdopodobne	2 punkty
niezbyt prawdopodobne	3 punkty
umiarkowanie prawdopodobne	4 punkty
prawdopodobne	5 punktów
bardzo prawdopodobne	6 punktów

Zsumuj punkty dla obszaru 2 (dla pytań 7-12) i podziel przez 6.

Pytania związane z obszarem 3 – asertywność

Pomyśl o sytuacji, w której dzielisz się pomysłem, sądząc że może on się nie spodobać Twoim koleżankom i kolegom.

13. Jak dobrze lub źle byś sobie z tym radził w swoim zespole?	
bardzo źle	1 punkt
źle	2 punkty
raczej źle	3 punkty
umiarkowanie dobrze	4 punkty
dobrze	5 punktów
bardzo dobrze	6 punktów

14. Jak komfortowo lub niekomfortowo czułbyś się, wykonując to w swoim zespole?	
bardzo niekomfortowo	1 punkt
niekomfortowo	2 punkty
niezbyt niekomfortowo	3 punkty
umiarkowanie komfortowo	4 punkty
komfortowo	5 punktów
bardzo komfortowo	6 punktów



15. Jak bardzo prawdopodobne lub mało prawdopodobne byłoby, abyś wykonał to w swoim zespole?	
bardzo mało prawdopodobne	1 punkt
mało prawdopodobne	2 punkty
niezbyt prawdopodobne	3 punkty
umiarkowanie prawdopodobne	4 punkty
prawdopodobne	5 punktów
bardzo prawdopodobne	6 punktów

Pomyśl o sytuacji, w której przedstawiasz pomysł, który różni się od pomysłu, o którym właśnie skończyliście dyskutować.

16. Jak dobrze lub źle byś sobie z tym radził w swoim zespole?	
bardzo źle	1 punkt
źle	2 punkty
raczej źle	3 punkty
umiarkowanie dobrze	4 punkty
dobrze	5 punktów
bardzo dobrze	6 punktów



17. Jak komfortowo lub niekomfortowo czułbyś się, wykonując to w swoim zespole?	
bardzo niekomfortowo	1 punkt
niekomfortowo	2 punkty
niezbyt niekomfortowo	3 punkty
umiarkowanie komfortowo	4 punkty
komfortowo	5 punktów
bardzo komfortowo	6 punktów

18. Jak bardzo prawdopodobne lub mało prawdopodobne byłoby, abyś wykonał to w swoim zespole?	
bardzo mało prawdopodobne	1 punkt
mało prawdopodobne	2 punkty
niezbyt prawdopodobne	3 punkty
umiarkowanie prawdopodobne	4 punkty
prawdopodobne	5 punktów
bardzo prawdopodobne	6 punktów

Zsumuj punkty dla obszaru 3 (dla pytań 13-18) i podziel przez 6.

Pytania związane z obszarem 4 – otwartość i uważność w zadawaniu pytań

Pomyśl o sytuacji, w której prosisz koleżankę/kolegę o wyjaśnienie ich pomysłu w inny sposób, żebyś lepiej mógł go zrozumieć.

19. Jak dobrze lub źle byś sobie z tym radził w swoim zespole?	
bardzo źle	1 punkt
źle	2 punkty
raczej źle	3 punkty
umiarkowanie dobrze	4 punkty
dobrze	5 punktów
bardzo dobrze	6 punktów

20. Jak komfortowo lub niekomfortowo czułbyś się, wykonując to w swoim zespole?	
bardzo niekomfortowo	1 punkt
niekomfortowo	2 punkty
niezbyt niekomfortowo	3 punkty
umiarkowanie komfortowo	4 punkty
komfortowo	5 punktów
bardzo komfortowo	6 punktów



21. Jak bardzo prawdopodobne lub mało prawdopodobne byłoby, abyś wykonał to w swoim zespole?	
bardzo mało prawdopodobne	1 punkt
mało prawdopodobne	2 punkty
niezbyt prawdopodobne	3 punkty
umiarkowanie prawdopodobne	4 punkty
prawdopodobne	5 punktów
bardzo prawdopodobne	6 punktów

Pomyśl o sytuacji, w której prosisz koleżankę/kolegę o powtórzenie, bo nie jesteś pewien, czy zrozumiałeś właściwie jej/jego pomysł.

22. Jak dobrze lub źle byś sobie z tym radził w swoim zespole?	
bardzo źle	1 punkt
źle	2 punkty
raczej źle	3 punkty
umiarkowanie dobrze	4 punkty
dobrze	5 punktów
bardzo dobrze	6 punktów



23. Jak komfortowo lub niekomfortowo czułbyś się, wykonując to w swoim zespole?	
bardzo niekomfortowo	1 punkt
niekomfortowo	2 punkty
niezbyt niekomfortowo	3 punkty
umiarkowanie komfortowo	4 punkty
komfortowo	5 punktów
bardzo komfortowo	6 punktów

24. Jak bardzo prawdopodobne lub mało prawdopodobne byłoby, abyś wykonał to w swoim zespole?	
bardzo mało prawdopodobne	1 punkt
mało prawdopodobne	2 punkty
niezbyt prawdopodobne	3 punkty
umiarkowanie prawdopodobne	4 punkty
prawdopodobne	5 punktów
bardzo prawdopodobne	6 punktów

Zsumuj punkty dla obszaru 4 (dla pytań 19-24) i podziel przez 6.

Pytania związane z obszarem 5 – koncentracja i uważne słuchanie

Pomyśl o tym, jak łatwo lub jak trudno byłoby Tobie wykonać to, co jest opisane poniżej.

25. Słuchaj uważnie koleżanki/kolegi z zespołu, która/-y dzieli się swoim pomysłem, zamiast koncentrować się na tym, co chcesz powiedzieć zespołowi o Twoim pomysłu.	
bardzo trudno	1 punkt
trudno	2 punkty
niezbyt trudno	3 punkty
umiarkowanie łatwo	4 punkty
łatwo	5 punktów
bardzo łatwo	6 punktów

26. Utrzymuj koncentrację na rozmowie z Twoim zespołem, nie odpływaj myślami.	
bardzo trudno	1 punkt
trudno	2 punkty
niezbyt trudno	3 punkty
umiarkowanie łatwo	4 punkty
łatwo	5 punktów
bardzo łatwo	6 punktów



27. W pełni skoncentruj się na tym, co mówi koleżanka/kolega z zespołu, zamiast myśleć o tym, co za chwilę sam powiesz zespołowi.	
bardzo trudno	1 punkt
trudno	2 punkty
niezbyt trudno	3 punkty
umiarkowanie łatwo	4 punkty
łatwo	5 punktów
bardzo łatwo	6 punktów

28. Koncentruj się na tym, co mówi koleżanka/kolega z zespołu, podczas, gdy sam wolałbyś pracować nad swoją częścią zadania zespołowego.	
bardzo trudno	1 punkt
trudno	2 punkty
niezbyt trudno	3 punkty
umiarkowanie łatwo	4 punkty
łatwo	5 punktów
bardzo łatwo	6 punktów

Zsumuj punkty dla obszaru 5 (dla pytań 25-28) i podziel przez 4.



SCENARIUSZE POSZCZEGÓLYCH ZAJĘĆ

ZAJĘCIA 0.

TEMAT: WPROWADZENIE W PROJEKT

CEL OGÓLNY ZAJĘĆ:

Zapoznanie z planem pracy podczas pełnego cyklu zajęć.

CELE SZCZEGÓŁOWE:

Uczeń:

- wie, w jaki sposób będzie przebiegać praca w trakcie cyklu zajęć,
- zna podstawowe zasady pracy zespołowej,
- potrafi dokonać samooceny i ma świadomość swoich kompetencji w zakresie komunikacji i współpracy w zespole.

METODY I FORMY PRACY:

- **ogładowe i słowne** – nauczyciel objaśnia cel główny cyklu zajęć, jego zakres tematyczny, wskazuje na powiązanie z podstawą programową, omawia materiały, które są do dyspozycji uczniów,
- **praktyczne** – uczniowie wykonują testy, do których dostęp uzyskali od nauczyciela, w ten sposób poddają refleksji sytuacje i stany opisane w ww. testach i dokonują wyboru odpowiedzi najbardziej odpowiadających ich samoocenie,
- **aktywizujące** – uczniowie rozmawiają nt. znajomości zagadnień omawianych podczas zajęć, w tym swoich wyobrażeń lub doświadczeń dotyczących programowania (w tym w języku Python), jak i pozostałych kwestii objętych cyklem zajęć (HTLM, grafika etc.).

ŚRODKI DYDAKTYCZNE:

- Nauczyciel zapoznaje się z Materiałami dla nauczycieli.

- Nauczyciel zapoznaje się z treścią przygotowanych dla uczniów filmów dotyczących komunikacji i współpracy w zespole. Pozwoli mu to lepiej zaplanować zajęcia, ew. uzupełnić wiedzę i wejść w rolę tutora, wspierającego uczniów w realizacji projektów zespołowych.



<https://diode.zone/w/p/pDPMbrpHeN5Ks81V8MxQQB>

- Nauczyciel przygotowuje *Test umiejętności współpracy i komunikacji w zespole* oraz *Test predyspozycji i ról*. Szablony można pobrać z poniższych linków:
- test umiejętności współpracy – <https://tinyurl.com/32rpw7wf>
- test predyspozycji – <https://tinyurl.com/2v6kneeh>

Następnie należy zduplikować je i umieścić na swoim koncie Microsoft, wybrać przycisk “Udostępnij” i po wybraniu odpowiednich ustawień skopiować adresy poszczególnych formularzy i udostępnić je uczniom do wypełnienia bezpośrednio przed rozpoczęciem testu.

- Nauczyciel powinien zdecydować, czy za wykonane zadania będą przyznawane oceny (rekomendujemy motywowanie przez nagrodę, np. ocenę celującą za ukończenie projektu, tj. osiągnięcie opisanego powyżej celu zajęć w dwóch wymiarach, lub możliwość wykorzystania stworzonej aplikacji w ramach funkcjonowania szkoły lub klasy – zachęcamy, aby uczniowie poszukali takich tematów i funkcjonalności aplikacji, które mogą być przydatne innym uczniom i/lub nauczycielom).

PRZEBIEG ZAJĘĆ:

Nauczyciel przedstawia uczniom cele zajęć oraz sposób, w jaki będą realizowane:

- celami zajęć są: wspólna praca nad projektem, poznanie języka programowania, narzędzi edycyjnych etc.,

- uczniowie będą pracować nad stworzeniem prostej aplikacji, ucząc się przy tym języka programowania Python oraz pozostałych narzędzi,
- uczniowie będą pracować w zespołach, na których skład nie mają wpływu, ponieważ odzwierciedla to prawdopodobne realia ich przyszłej pracy zawodowej,
- uczniowie mają do dyspozycji wspomagające materiały wideo i wsparcie nauczyciela.

Nauczyciel razem z uczniami opracowują zasady pracy zespołowej czyli kontrakt, w którym zawarte są zasady formalne (więcej informacji dot. zasad pracy w Przewodniku dla nauczycieli).

Nauczyciel rozmawia z uczniami o ich wiedzy i znajomości języków programowania, w szczególności języka Python, oraz innych zagadnień omawianych w trakcie zajęć. Wykorzystuje tę wiedzę podczas podziału uczniów na zespoły.

Następnie przeprowadza testy i archiwizuje wyniki:

- test umiejętności współpracy i komunikacji w zespole (ok 15 minut).
- test predyspozycji i ról (ok. 5 minut). Wyniki posłużą do dokonania podziału uczniów na zespoły.

Informuje uczniów, w jakim terminie poznają skład zespołów. Termin powinien być jak najkrótszy, aby uczniowie mogli nawiązać kontakt przed zajęciami nr 1.

Zachęca uczniów do:

- obejrzenia w czasie pozalekcyjnym filmów dotyczących komunikacji i pracy w zespole i zapoznania się z Przewodnikiem dla uczniów;
- niezwłocznego kontaktu w ramach zespołu.

Jeżeli podczas zajęć ma przestrzeń czasową, ogląda razem z uczniami i omawia wybrane filmy z zakresu komunikacji i współpracy w zespole. Rekomendujemy na początek filmy nr 1, 2 i 4.

Po zakończeniu zajęć (lub w ich przerwie, jeśli zajęcia trwają dłużej niż 45 minut) nauczyciel:

- dokonuje podziału klasy na zespoły czteroosobowe. Jeśli to konieczne ze względu na arytmetykę, jeden z zespołów może liczyć od 2 do 5 osób (więcej informacji o zespołach i podziale w Przewodniku dla nauczycieli),
- w optymalnej sytuacji każdy członek zespołu reprezentuje inny typ predyspozycji (w teście – kolor); w przypadku nierównomiernego rozkładu typów predyspozycji w klasie bierzemy pod uwagę drugi wybór,
- należy uwzględnić parytet płci; optymalnie jest, gdy w zespole są dwie dziewczyny i dwóch chłopców; jeśli nie ma takiej możliwości, to dążymy do jednorodności zespołu pod względem płci,
- należy uwzględnić poziom wiedzy i zaawansowania, jeśli chodzi o zagadnienia programistyczne lub znajomość pozostałych umiejętności merytorycznych wymaganych podczas zajęć (język HTML, wykorzystanie programów graficznych, tworzenie dokumentacji); dbamy więc, aby w jednym zespole nie znalazły się wyłącznie osoby najbardziej lub najmniej zaawansowane,
- należy zadbać, aby w jednym zespole nie znalazły się osoby silnie dominujące lub jawnie ze sobą skonfliktowane.

Informację o składach zespołów nauczyciel przekazuje uczniom, co najmniej na 2 dni przed zajęciami 1 (za wyjątkiem, gdy podział zostanie dokonany w przerwie między jednostkami lekcyjnymi).

WSKAZÓWKI METODYCZNE:

Zajęcia mogą być realizowane w oparciu o 3 przykładowe projekty:

Lokalizacje lotów samolotów: <https://aviationstack.com> (w filmach używamy tego projektu).

Weryfikacja numerów telefonów z podaniem danych o operatorze i kraju: <https://numverify.com>

Alert pogodowy dla 3-dniowej prognozy: <https://wttr.in/>

Każdy zespół może wybrać dowolny inny projekt (inne API, inne funkcjonalności); ważne, aby zachować układ: aplikacja, dokumentacja, strona internetowa o produkcie. Jeśli nauczyciel uzna za stosowne, aby ułatwić uczniom zadanie, może zasugerować im wybranie projektu dot. lokalizacji lotów samolotów – będą wówczas odtwarzać przykłady zamieszczone w filmach instruktażowych. Z punktu widzenia celu zajęć ważne jest przeprowadzenie procesu budowania aplikacji, aby młodzież nauczyła się współpracy w zespole. Najbardziej ambitne zespoły mogą, za zgodą nauczyciela, dowolnie kształtować wykonywane przez siebie projekty. Nauczyciel nie powinien ograniczać ich inwencji nawet jeśli ma poczucie, że mogą przewyższać go wiedzą techniczną.

Jeśli nauczyciel informatyki nie czuje się kompetentny w obszarze kompetencji społecznych, aby sam odtwarzał i omawiał z uczniami filmy dot. współpracy i komunikacji w zespole, zachęcamy, aby podjął współpracę z wychowawcą klasy (jeśli zajęcia są realizowane w ramach lekcji) i poprosił go/ją o przeznaczenie 2 jednostek lekcyjnych na odtworzenie i omówienie wszystkich 10 filmów z tej serii. Z uwagi na nieliniarny proces tworzenia i docierania się zespołów projektowych, zalecamy odtworzenie i omówienie filmów na wczesnym etapie realizacji cyklu (aby odpowiednio wcześniej przekazać uczniom praktyczne wskazówki dot. komunikacji i współpracy oraz radzenia sobie z trudnościami w tym zakresie)

MATERIAŁY POMOCNICZE:

Uczniom, którzy chcieliby samodzielnie poznać podstawy języka, nauczyciel sugeruje, aby skorzystali z bezpłatnych szkoleń dostępnych w portalu OSE:

- podstawowy: <https://it-szkola.edu.pl/kkurs,kurs,216> wraz z kodami w serwisie GitHub: https://github.com/klubmlodegoprogramisty/python/tree/main/poziom_podstawowy
- średnio zaawansowany: <https://it-szkola.edu.pl/kkurs,kurs,217> wraz z kodami w serwisie GitHub: https://github.com/klubmlodegoprogramisty/python/tree/main/poziom_sredniozaawansowany

lub wskazuje inne źródła, które uznaje za wartościowe.

ZAJĘCIA 1.

TEMAT: PRZYGOTOWANIE ŚRODOWISKA – KONTA, OPROGRAMOWANIE

CEL OGÓLNY ZAJĘĆ:

Nauka samodzielnego przygotowania środowiska programistycznego i środowiska do zespołowej pracy programistycznej, zdobycie wiedzy nt. licencji stosowanych w pracy programisty.

Integracja uczniów w zespołach. Rozwijanie otwartości na współpracę z innymi członkami zespołu.

CELE SZCZEGÓŁOWE:

Uczeń:

- zna podstawowe zasady pracy w zespole, do którego został przydzielony,
- zna podstawowe zasady funkcjonowania środowiska GitHub,
- potrafi założyć konto w serwisie GitHub oraz zmienić jego ustawienia,
- potrafi samodzielnie skonfigurować wirtualne środowisko dla pojedynczego lokalnego projektu w oparciu o przedstawiony schemat,
- identyfikuje się z zasadami współpracy wypracowanymi wspólnie z koleżankami i kolegami z zespołu.
- podejmuje indywidualne i zespołowe decyzje.

METODY I FORMY PRACY:

- **oglądowe i słowne** – nauczyciel, po krótkim wprowadzeniu odtworza i ewentualnie omawia filmy instruktażowe, zawierające prezentację czynności wykonywanych przez autora filmu;
- **praktyczne** – uczniowie indywidualnie lub w zespole podejmują

się wykonania operacji zaprezentowanych w filmach, adaptując je do zastosowania w opracowywanym rozwiązaniu (docelowej aplikacji);

- **aktywizujące** – uczniowie, w ramach utworzonych zespołów, dyskutują nt. przyjęcia optymalnych rozwiązań w celu realizacji zaplanowanego projektu.

ŚRODKI DYDAKTYCZNE:

- Przed rozpoczęciem zajęć należy zweryfikować działanie VirtualBox na komputerach szkolnych – to konieczne w celu powodzenia zajęć.
- Przed zajęciami nauczyciel zapoznaje się z filmami, aby potrafił odpowiedzieć na pytania uczniów; w szczególności zalecamy film A02, A03, A04.



<https://diode.zone/w/p/cKkiLF9zUFjVTQhJL5iiPf>

- Filmy posiadają dwa oznaczenia początkowe:
 - oznaczenie „A” posiadają filmy dedykowane nauce języka Python,
 - oznaczenie „B” posiadają filmy przekazujące wiedzę i umiejętności okołoprogramistyczne, pozwalające na kompletne wykonanie aplikacji (HTML, grafika, edytor tekstu).

Ta struktura dotyczy wszystkich filmów opracowanych na potrzeby cyklu zajęć.

- Można wskazać na różne rodzaje licencji różnych systemów operacyjnych i samego Pythona w kontekście prawa autorskiego.

PRZEBIEG ZAJĘĆ:

Zajęcia rozpoczynamy od spotkania uczniów w zespołach, do których zostali przydzieleni. Uczniowie samodzielnie wypracowują kontrakt regulujący zasady współpracy w ich zespole (więcej informacji w Przewodniku dla nauczycieli), a także mogą nadać nazwę swoim zespołom. Celem jest stworzenie poczucia przynależności i integracji zespołu. Uczniowie mogą też zacząć zastanawiać się nad tematem aplikacji, nad którą będą pracować.

Następnie rozpoznajemy wiedzę uczniów. Warto podyskutować chwilę o ich znajomości serwisu GitHub i o tym, jak wykorzystują go firmy do wspólnej pracy programistów.

Po takim wstępie nauczyciel prezentuje filmy i komentuje je w tych momentach, które uzna za stosowne w zależności od wiedzy uczniów. Uczniowie oglądają i starają się wykonać te elementy, które potrafią, ewentualnie przy wsparciu nauczyciela wykonują je po każdym filmie. Poniżej lista filmów:

- Konfiguracja IDE PyCharm, tworzenie konta w [GitHub.com](https://github.com) (A01)
Z uwagi na fakt, że w OVA w VirtualBox wszystko już jest zainstalowane, uczniowie tylko zakładają sobie konta w serwisie [GitHub.com](https://github.com). Tę czynność można zlecić do wykonania samodzielnie w domu przed zajęciami.
- Przygotowujemy środowisko venv dla lokalnego projektu (A02)
- Plik `requirements.txt` – zewnętrzne moduły, własne pliki py w projekcie (A03)
- Minimalny program z wykorzystaniem PySimpleGUI (A04)
- Skąd będziemy czerpać grafiki? Pixabay, Freepik i Flaticon (B01)

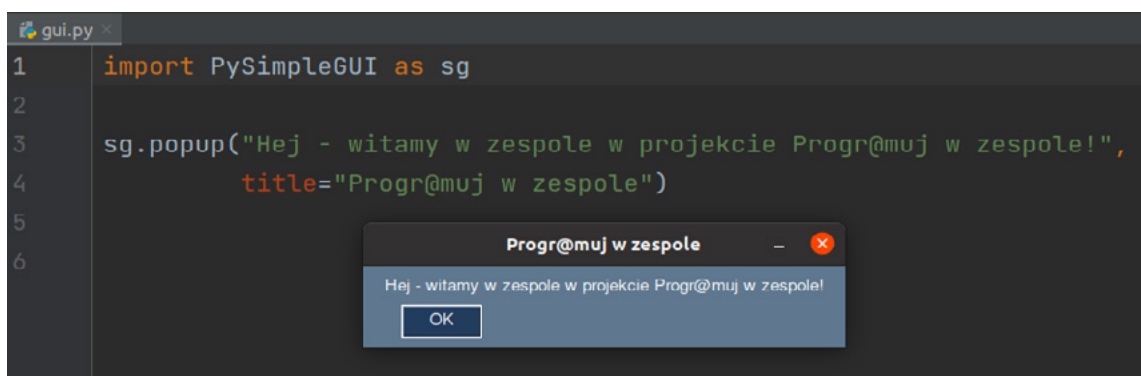


<https://creativecommons.pl/wyberz-licencje/>

- Uruchamiamy edytor tekstów – format strony (B02)
- Uruchamiamy edytor grafiki – otwieramy przykładowy plik graficzny (B03)
- Uruchamiamy edytor HTML – podstawowa strona z szablonu (B04)
- Edytor HTML: Bootstrap – <https://getbootstrap.com/> (B05)

WSKAZÓWKI METODYCZNE:

- Tuż po filmie A03 uczniowie mogą próbować uruchomić aplikację DEMO To pozwoli im na wczesnym etapie zobaczyć, jak może wyglądać efekt końcowy.
- W części A03 można pokazać stronę <https://pypi.org/> (to repozytorium zewnętrznych bibliotek do Pythona, np.: <https://pypi.org/project/PyTechBrain/>)
- Aplikacja wykonana przy części A04 powinna wyglądać następująco:



Kod źródłowy:

```
import PySimpleGUI as sg

sg.popup("Hej – witamy w zespole w projekcie Progr@muj w zespole!",
        title="Progr@muj w zespole")
```

- W częściach B01, B02, B03, B04 można przypomnieć o licencjach, zgodnie z zapisami podstawy programowej uczniowie mają rozróżniać różne licencje.
- W części B05 dodatkowo można przypomnieć jak wiele oprogramowania jest tworzonego na otwartych licencjach, np.:



<https://github.com/twbs/bootstrap/blob/v5.0.2/LICENSE>

- Zajęcia zostały zaplanowane do przeprowadzenia w czasie jednej jednostki lekcyjnej (45 minut). Jeśli jednak nauczyciel – ze względu na tempo pracy zespołów lub złożoność zagadnienia – uzna, że to za mało, zalecamy rozważenie takiej organizacji zajęć, aby można było dokończyć je na kolejnej jednostce lekcyjnej (dodatkowe do 45 minut).

MATERIAŁY POMOCNICZE:

W przypadku problemów z pobieraniem pakietów (bardzo częste dla połączeń **OSE**), zalecamy umieszczenie pliku `pip.conf` w katalogu użytkownika: `$HOME/.pip/pip.conf` o zawartości:

```
[global]
trusted-host = pypi.python.org
               pypi.org
               files.pythonhosted.org
```

- Więcej informacji o konfiguracji narzędzia pip:
<https://pip.pypa.io/en/stable/topics/configuration/>
- Strona opisująca różne rodzaje licencji CC:
<https://creativecommons.pl/wyberz-licencje/>
- Strony opisująca różne rodzaje licencji GPL:



<https://opensource.org/licenses>

- <https://license.md/free-versus-open-source-license-what-is-the-difference-everything-you-need-to-know/>
- <https://itsfoss.com/open-source-licenses-explained/>

ZAJĘCIA 2.

TEMAT: VIRTUALBOX I POZOSTAŁE APLIKACJE

CEL OGÓLNY ZAJĘĆ:

Nauka wykorzystania wirtualnego środowiska komputerowego wewnątrz innego systemu operacyjnego oraz podstawowych operacji w edytorach.

Rozwijanie otwartości na współpracę z innymi członkami zespołu.

CELE SZCZEGÓŁOWE:

Uczeń:

- zna podstawowe zasady funkcjonowania systemu VirtualBox,
- potrafi zidentyfikować znaczniki meta w dokumencie HTML,
- potrafi tworzyć dokumenty w edytorze tekstów,
- potrafi wyróżnić bitmapowe (rastrowe) formaty graficzne,
- współpracuje z innymi członkami zespołu podczas wykonywania operacji zaprezentowanych w filmach instruktażowych,
- komunikuje się w sposób klarowny podczas dyskusji z kolegami i koleżankami w zespole,
- słucha z uwagą i szanuje wypowiedzi innych.

METODY I FORMY PRACY:

- **ogładowe i słowne** - nauczyciel, po krótkim wprowadzeniu, odtworza i ewentualnie omawia filmy instruktażowe, zawierające prezentację czynności wykonywanych przez autora filmu,
- **praktyczne** - uczniowie indywidualnie lub w zespole podejmują się wykonania operacji zaprezentowanych w filmach, adaptując je do zastosowa-

nia w opracowywanym rozwiązaniu,

- **aktywizujące** - uczniowie, w ramach utworzonych zespołów, dyskutują nt. przyjęcia optymalnych rozwiązań w celu realizacji zaplanowanego projektu.

ŚRODKI DYDAKTYCZNE

- Przed rozpoczęciem zajęć należy zweryfikować działanie VirtualBox na komputerach szkolnych - to konieczne w celu powodzenia zajęć.
- Przed zajęciami nauczyciel zapoznaje się z filmami, aby potrafił odpowiedzieć na pytania uczniów; w szczególności zalecamy film B06.



<https://diode.zone/w/p/hmgHJGMCCQeYhN9obfJnyn>

- Można wskazać na różne rodzaje licencji różnych systemów operacyjnych w kontekście prawa autorskiego i używania VirtualBox.

PRZEBIEG ZAJĘĆ:

Nauczyciel prezentuje filmy i komentuje je w tych momentach, które uzna za stosowne w zależności od wiedzy uczniów. Uczniowie oglądają, starają się wykonać te elementy, które potrafią, ewentualnie przy wsparciu nauczyciela wykonują je po każdym filmie. Poniżej lista filmów:

- VirtualBox w Windows i jak importować maszynę OVA - aby pracować niezależnie od szkoły (B06)
- PySimpleGui - dokumentacja, przykłady użycia (A05)

Warto zwrócić uwagę na sposób tworzenia:



<https://pysimplegui.readthedocs.io/en/latest/#open-source-license-but-private-development>

- Edytor tekstów: nagłówek i stopka (B07)
- Edytor tekstów: style i spis treści (B08)
- Edytor grafiki: zmiana rozmiaru i zapis XCF (B09)
- Edytor HTML: różne znaczniki meta (B10)

WSKAZÓWKI METODYCZNE

- W częściach B07, B08, B09, B10 można przypomnieć o licencjach, zgodnie z zapisami podstawy programowej uczniowie mają rozróżniać różne licencje.
- W części B10 dodatkowo można przypomnieć jak wiele oprogramowania jest tworzonego na otwartych licencjach, np.: <https://github.com/twbs/bootstrap/blob/v5.0.2/LICENSE>.
- Zajęcia zostały zaplanowane do przeprowadzenia w czasie jednej jednostki lekcyjnej (45minut). Jeśli jednak nauczyciel - ze względu na tempo pracy zespołów lub złożoność zagadnienia - uzna, że to za mało, zalecamy rozważenie takiej organizacji zajęć, aby można było dokończyć je na kolejnej jednostce lekcyjnej (dodatkowe do 45 minut).

MATERIAŁY POMOCNICZE

- Strona opisująca różne rodzaje licencji CC:
 - <https://creativecommons.pl/wyberz-licencje/>

- Strony opisująca różne rodzaje licencji GPL:



<https://opensource.org/licenses>

- <https://license.md/free-versus-open-source-license-what-is-the-difference-everything-you-need-to-know/>
- <https://itsfoss.com/open-source-licenses-explained/>

ZAJĘCIA 3.

TEMAT: PYTHON - PODSTAWY, DANE, PĘTLE, INSTRUKCJE WARUNKOWE

CEL OGÓLNY ZAJĘĆ:

Nauka podstaw języka Python, podstawowe typy danych, konstrukcje programistyczne pętli iteracyjnej oraz instrukcji warunkowej. Te elementy są ściśle powiązane z podstawą programową szkoły ponadpodstawowej.

Rozwijanie otwartości na współpracę z innymi członkami zespołu.

CELE SZCZEGÓŁOWE:

Uczeń:

- zna podstawowe typy danych w Python,
- potrafi stosować pętle iteracyjne w połączeniu z kolekcjami danych,
- używa konstrukcji instrukcji warunkowej w Python,
- współpracuje z innymi członkami zespołu podczas wykonywania operacji zaprezentowanych w filmach instruktażowych,
- komunikuje się w sposób klarowny podczas dyskusji z kolegami i koleżankami w zespole.
- słucha z uwagą i szanuje wypowiedzi innych.

METODY I FORMY PRACY:

- **oglądowe i słowne** - nauczyciel, po krótkim wprowadzeniu, odtworza i ewentualnie omawia filmy instruktażowe, zawierające prezentację czynności wykonywanych przez autora filmu;
- **praktyczne** - uczniowie indywidualnie lub w zespole podejmują

się wykonania operacji zaprezentowanych w filmach, adaptując je do zastosowania w opracowywanym rozwiązaniu (docelowej aplikacji);

ŚRODKI DYDAKTYCZNE

- Przed rozpoczęciem zajęć należy zweryfikować działanie VirtualBox na komputerach szkolnych - to konieczne w celu powodzenia zajęć.
- Przed zajęciami nauczyciel zapoznaje się z filmami, aby potrafił odpowiedzieć na pytania uczniów.



<https://diode.zone/w/p/b1G5ys14WsSjjmnAHeJLGa>

Zalecamy przetestowanie kodów źródłowych zawartych w serwisie GitHub, aby sprawnie pomagać młodzieży: https://github.com/programujemy-python/programuj-w-zespole/tree/main/teachers_sample_codes/zajecia_03

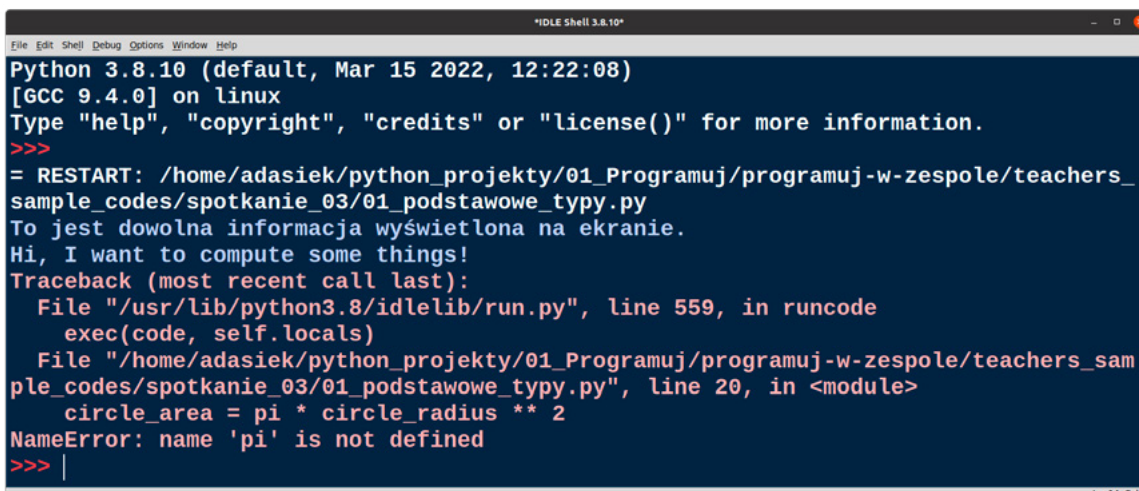
PRZEBIEG ZAJĘĆ:

Nauczyciel prezentuje filmy, komentując je w tych momentach, które uzna za stosowne - w zależności od wiedzy uczniów. Uczniowie oglądają, starają się wykonać w ich trakcie te elementy, które potrafią, ewentualnie przy wsparciu nauczyciela wykonują je po każdym filmie. Poniżej lista filmów:

- Podstawowe typy danych w Python, zmienne (A06)
- Typy zaawansowane: listy, słowniki (A07)
- Importowanie z zewnętrznych modułów (A08)
- Pętla `for` i listy (A09)
- Instrukcja warunkowa `if ... else ...` (A10)

WSKAZÓWKI METODYCZNE

- Zmienne zawierają wartości różnego typu i to na podstawie analizy tych wartości interpreter Pythona określa typ zmiennej. Nie deklarujemy typu zmiennej, musimy pamiętać, że typ zmiennej zmienia się z jej zawartością. Musimy również pamiętać, że w języku Python rozróżniamy wielkość liter: zmienna `linux` to coś innego niż `Linux`, a funkcja `Print()` to nie jest `print()`.
- W przypadku słowników zwróćmy uwagę na znaki dwukropka rozdzielające klucz i wartość.
- Pośrodku filmu A06 ujawnia się błąd! To jest zamierzone — aby pokazać, w jaki sposób Python komunikuje problemy i błędy w kodzie. W serwisie GitHub pozostawiamy kod z błędem, który uczniowie muszą samodzielnie poprawić.



```
Python 3.8.10 (default, Mar 15 2022, 12:22:08)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: /home/adasiak/python_projekty/01_Programuj/programuj-w-zespole/teachers_
sample_codes/spotkanie_03/01_podstawowe_typy.py
To jest dowolna informacja wyświetlona na ekranie.
Hi, I want to compute some things!
Traceback (most recent call last):
  File "/usr/lib/python3.8/idlelib/run.py", line 559, in runcode
    exec(code, self.locals)
  File "/home/adasiak/python_projekty/01_Programuj/programuj-w-zespole/teachers_sam
ple_codes/spotkanie_03/01_podstawowe_typy.py", line 20, in <module>
    circle_area = pi * circle_radius ** 2
NameError: name 'pi' is not defined
>>> |
```

Uwaga! To jest zdjęcie poglądowe z wykonania kodu w programie IDLE

Wprowadzamy tutaj od razu konwencję `F-String`, która jest obecnie (rok 2022) najpopularniejsza w Pythonie.

Przykład krótkiego wywołania `F-String` (pamiętamy o literze `f` przed znakami cudzysłowów):

```
some_value = 3.1415
some_text = f"PI Value is: {some_value}"
print(some_text)
# inny sposób:
print(f"PI Value is: {some_value}")
```

Słowniki - w Pythonie od wersji 3.6 słowniki zachowują pewne uporządkowanie, lecz traktujemy je jako nieuporządkowane, a więc niezachowujące kolejności występowania elementów

Ważne informacje dotyczące **importowania modułów**:

- `import modul` - w ten sposób wczytujemy cały moduł, a do jego poszczególnych elementów odwołujemy się poprzez zapis: `modul.funkcja()`
- `from modul import *` - w ten sposób wczytujemy wszystkie elementy z modułu, a odwołujemy się poprzez zapis `funkcja()`, bez podawania jawnie nazwy modułu
- `from modul import funkcja_a` - w ten sposób wczytujemy tylko pewną konkretną funkcję, a odwołujemy się poprzez zapis `funkcja_a()`, bez podawania jawnie nazwy modułu
- `from modul import funkcja_a as funkcja_b` - w ten sposób wczytujemy tylko pewną konkretną funkcję i nadajemy jej własną nazwę, a odwołujemy się poprzez zapis `funkcja_b()`, bez podawania jawnie nazwy modułu

Najczęściej stosujemy zapis trzeci - `from modul import funkcja_a`

- **Pętle** - na filmach w PyCharm są utworzone 2 pliki, podczas gdy w GitHub jest to jeden plik. To dla przejrzystości filmów. W szkole można wykonać to dowolnie — albo dwa pliki jak na filmie, albo jeden duży jak w GitHub. Pozostawiamy to decyzji nauczyciela.



Ciekawostka z serwisu *dot. pętli for Stackoverflow*.
<https://stackoverflow.com/questions/23262837/how-do-setup-virtualenv-and-dependencies-in-pycharm>

Instrukcje warunkowe - w języku Python używamy następującej składni (zapisu):

```
if warunek_1:
    # blok kodu gdy warunek_1 jest prawdą
elif warunek_2:
    # blok kodu gdy warunek_2 jest prawdą
elif warunek_3:
    # blok kodu gdy warunek_3 jest prawdą
elif warunek_n:
    # blok kodu gdy warunek_n jest prawdą
else:
    # blok kodu wykonywany wówczas, kiedy żaden z powyższych
    warunków nie jest spełniony
```

- **Odwołanie do urllib** - funkcje z modułu standardowego <https://docs.python.org/3/library/urllib.request.html#urllib.request.urlopen> są trudne w użyciu, dlatego powstała biblioteka:

Requests: HTTP for Humans™

(<https://requests.readthedocs.io/en/latest/>) - będziemy do niej wracać w kolejnych materiałach.

- Zajęcia zostały zaplanowane do przeprowadzenia w czasie jednej jednostki lekcyjnej (45 minut). Jeśli jednak nauczyciel - ze względu na tempo pracy zespołów lub złożoność zagadnienia - uzna, że to za mało, zalecamy rozważenie takiej organizacji zajęć, aby można było dokończyć je na kolejnej jednostce lekcyjnej (dodatkowe do 45 minut).

Pamiętamy o blokach kodu — wcięcie 4 spacje!

UWAGA!

W tym momencie zespoły mogą już wybierać temat aplikacji; w przykładowych są tylko 3 - być może grupa wybierze zupełnie inne API?

Zadanie domowe:

Dla utrwalenia wiedzy z tych zajęć możemy wykonać ćwiczenia:

1. Program, który bazując na roku urodzenia obliczy, w którym roku



<https://docs.python.org/3/library/datetime.html#datetime-objects>

dana osoba będzie mieć 100 lat oraz za ile to będzie lat. (wykorzystujemy tu standardową bibliotekę Pythona `datetime`)

2. Program, który sprawdzi, czy elementem kolekcji jest książka i poda jej tytuł. Mamy dwie kolekcje:

- listę `list` - zawierającą tuple (do których odwołujemy się jak do elementów listy),
- słownik `dict` - zawierający klucze (rodzaj) i wartości (tytuł).

Kod źródłowy znajduje się w pliku:
`teachers_sample_excercises/03_data.py`

MATERIAŁY POMOCNICZE

- Strona dokumentacji Python:
- <https://docs.python.org/3/>
- Materiały kursu OSE:
- <https://github.com/klubmlodegoprogramisty/python/>
- Materiały z książki
- https://github.com/abixadamj/helion-python/tree/main/Rozdzial_1



Materiały w serwisie Tik-Tok

- [Jak używać F-String](#)
- [Jak budować listę](#)
- [Pętla iteracyjna for](#)
- [Importowanie modułów](#)
- [Instrukcje warunkowe if ... else ...](#)

ZAJĘCIA 4.

TEMAT: PODSTAWY EDYTORÓW I PĘTLA

`while` W PYTHON

CEL OGÓLNY ZAJĘĆ:

Poznanie ważnych funkcjonalności edytorów, elementów `containers` języka HTML. Sposoby wyświetlania informacji oraz sterowanie zdarzeniami w bibliotece `PySimpleGUI`.

Rozwijanie otwartości na współpracę z innymi członkami zespołu.

CELE SZCZEGÓŁOWE:

Uczeń:

- zna elementy `containers` dokumentu HTML,
- potrafi użyć listy numerowanej lub nienumerowanej w edytorze tekstu,
- wyróżnia warstwy w edytorze grafiki i potrafi nimi manipulować,
- posługuje się biblioteką `PySimpleGUI`: wyświetla informacje tekstowe, obsługuje zdarzenia,
- potrafi wykorzystać pętlę nieskończoną `while True` i zna sposób jej przerywania,
- współpracuje z innymi członkami zespołu podczas wykonywania operacji zaprezentowanych w filmach instruktażowych,
- komunikuje się w sposób klarowny podczas dyskusji z kolegami i koleżankami w zespole.

METODY I FORMY PRACY:

- **ogładowe i słowne** - nauczyciel, po krótkim wprowadzeniu, od-

tworzą i ewentualnie omawia filmy instruktażowe, zawierające prezentację czynności wykonywanych przez autora filmu,

- **praktyczne** - uczniowie indywidualnie lub w zespole podejmują się wykonania operacji zaprezentowanych w filmach, adaptując je do zastosowania w opracowywanym rozwiązaniu,
- **aktywizujące** - uczniowie, w ramach utworzonych zespołów, dyskutują nt. przyjęcia optymalnych rozwiązań w celu realizacji zaplanowanego projektu.

ŚRODKI DYDAKTYCZNE

- Przed rozpoczęciem zajęć należy zweryfikować działanie VirtualBox na komputerach szkolnych - to konieczne w celu powodzenia zajęć.
- Przed zajęciami nauczyciel zapoznaje się z filmami, aby potrafił odpowiedzieć na pytania uczniów; w szczególności zalecamy filmy A12 i A13.



<https://diode.zone/w/p/6QKK8kEXhzvwQSDws3Msgt>

- Zalecamy przetestowanie kodów źródłowych zawartych w serwisie GitHub, aby sprawnie pomagać młodzieży: https://github.com/programujemy-python/programuj-w-zespole/tree/main/teachers_sample_codes/zajecia_04

PRZEBIEG ZAJĘĆ:

Nauczyciel prezentuje filmy i komentuje je w tych momentach, które uzna za stosowne w zależności od wiedzy uczniów. Uczniowie oglądają, starają się wykonać te elementy, które potrafią, ewentualnie przy wsparciu nauczyciela wykonują je po każdym filmie. Poniżej lista filmów:

- Edytor tekstów: listy numerowane i nienumerowane (B11)
- Edytor grafiki: warstwy i dodanie elementu (B12)



<https://gimpdlazielonych.pl/2019/05/31/warstwy-w-gimpie-podstawy/>

- Edytor HTML: containers (B13)



<https://getbootstrap.com/docs/5.0/layout/containers>

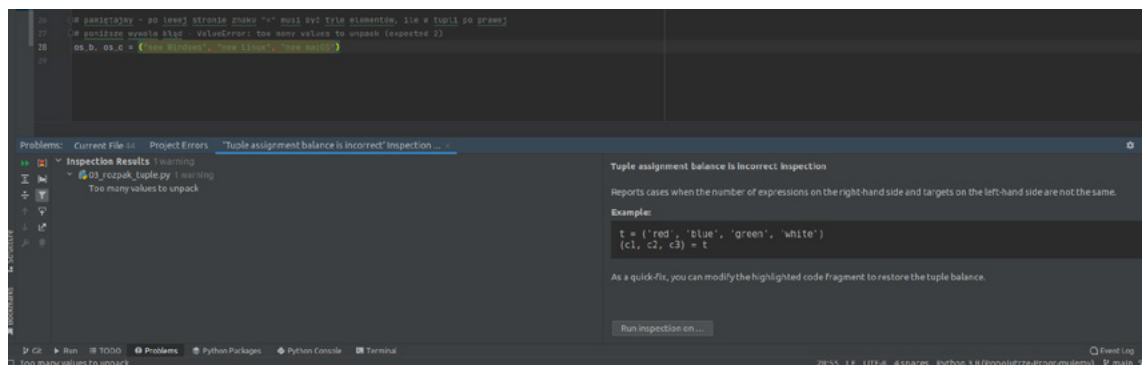
- Wyświetlamy informację. (A11)
- Rozpakowywanie tupli - pythonizm. (A12)
- Pętla `while True` - sterowanie programem PySimpleGUI. (A13)

WSKAZÓWKI METODYCZNE

- **Rozpakowywanie tupli** to tzw. pythonizm.

```
# pamiętajmy – po lewej stronie znaku "=" musi być tyle ele-  
mentów, ile w tupli po prawej  
# poniższe wywoła błąd – ValueError: too many values to  
unpack (expected 2)  
os_b, os_c = ("new Windows", "new Linux", "new macOS")
```





- Zajęcia zostały zaplanowane do przeprowadzenia w czasie jednej jednostki lekcyjnej (45 minut). Jeśli jednak nauczyciel - ze względu na tempo pracy zespołów lub złożoność zagadnienia - uzna, że to za mało, zalecamy rozważenie takiej organizacji zajęć, aby można było dokończyć je na kolejnej jednostce lekcyjnej (dodatkowe do 45 minut).

MATERIAŁY POMOCNICZE

Licencja grafiki bank.png: <https://www.freepikcompany.com/legal>



ZAJĘCIA 5.

TEMAT: ELEMENTY STERUJĄCE W PySimpleGUI

CEL OGÓLNY ZAJĘĆ:

Poznanie elementów wizualnych i sterujących aplikacji desktopowych opartych na bibliotece PySimpleGUI.

Rozwijanie otwartości na współpracę z innymi członkami zespołu.

CELE SZCZEGÓŁOWE:

Uczeń:

- tworzy aplikację zorientowaną na zdarzenia,
- potrafi definiować wygląd aplikacji desktopowej,
- zna sposoby wywoływania funkcji w zależności od zdarzenia,
- potrafi pobierać dane wpisane przez użytkownika,
- współpracuje z innymi członkami zespołu podczas wykonywania operacji zaprezentowanych w filmach instruktażowych,
- komunikuje się w sposób klarowny podczas dyskusji z kolegami i koleżankami w zespole.

METODY I FORMY PRACY:

- **ogładowe i słowne** - nauczyciel, po krótkim wprowadzeniu, odtworza i ewentualnie omawia filmy instruktażowe, zawierające prezentację czynności wykonywanych przez autora filmu,
- **praktyczne** - uczniowie indywidualnie lub w zespole podejmują się wykonania operacji zaprezentowanych w filmach, adaptując je do zastosowania w opracowywanym rozwiązaniu,



- **aktywizujące** - uczniowie, w ramach utworzonych zespołów, dyskutują nt. przyjęcia optymalnych rozwiązań w celu realizacji zaplanowanego projektu.

ŚRODKI DYDAKTYCZNE

- Przed rozpoczęciem zajęć należy zweryfikować działanie VirtualBox na komputerach szkolnych - to konieczne w celu powodzenia zajęć.
- Przed zajęciami nauczyciel zapoznaje się z filmami, aby potrafił odpowiedzieć na pytania uczniów.



<https://diode.zone/w/p/8xEUrJg54U1c4QbumLz356>

- Zalecamy przetestowanie kodów źródłowych zawartych w serwisie GitHub, aby sprawnie pomagać młodzieży: https://github.com/programujemy-python/programuj-w-zespole/tree/main/teachers_sample_codes/zajecia_05

PRZEBIEG ZAJĘĆ:

Nauczyciel prezentuje filmy i komentuje je w tych momentach, które uzna za stosowne w zależności od wiedzy uczniów. Uczniowie oglądają, starają się wykonać te elementy, które potrafią, ewentualnie przy wsparciu nauczyciela wykonują je po każdym filmie. Poniżej lista filmów:

- Dodajemy elementy przycisków. (A14)
- Dodajemy wyświetlanie obrazków. (A15)
- Sposoby wprowadzania danych. (A16)



<https://tinyurl.com/prog-python-psg-input>

- Poznajemy sterowanie. (A17)
- PySimpleGui - tworzymy prosty program okienkowy - 1 (layout, listy) (A18)
- Wyświetlanie większej ilości danych (output) (A19)

WSKAZÓWKI METODYCZNE

- W przypadku użycia grafik wskazujemy na konieczność atrybucji (oznaczenia), np.:

```
# https://www.flaticon.com/free-icon/smartphone_1257230  
# attribution: 'image: Flaticon.com'. This cover has been  
designed using resources from Flaticon.com
```

- Dla elementów `sg.Input()` zwracamy uwagę na kolejność kluczy w słowniku, który jest zwracany w operacji `event, values = window.read()`; tu również przypominamy o *rozpakowywaniu tupli*. Ciekawostka - specyficzne wywołanie dla pobrania nazwy pliku lub katalogu:



<https://pysimplegui.readthedocs.io/en/latest/#popup-input>

- Zajęcia zostały zaplanowane do przeprowadzenia w czasie jednej jednostki lekcyjnej (45 minut). Jeśli jednak nauczyciel - ze względu na tempo pracy zespołów lub złożoność zagadnienia - uzna, że to za mało, zalecamy rozważenie takiej organizacji zajęć, aby można było dokończyć je na kolejnej jednostce lekcyjnej (dodatkowe do 45 minut).

MATERIAŁY POMOCNICZE



Strona <https://www.freepikcompany.com/legal> - licencja na grafiki.

ZAJĘCIA 6.

TEMAT: DALSZE DZIAŁANIA W EDYTORACH, MODUŁ `requests` W PYTHON CONSOLE

CEL OGÓLNY ZAJĘĆ:

Operacje w edytorach; zrzuty ekranowe, warstwy, obiekty w dokumentach HTML. Nauka wykonywania połączeń do systemu API, wykorzystując Python Console w PyCharm; wykorzystanie modułu `requests` do komunikacji z API.

Rozwijanie kompetencji wymiany informacji i dzielenia się wiedzą poprzez dyskusję.

CELE SZCZEGÓŁOWE:

Uczeń:

- zna podstawowe zasady wywołań modułu `requests`,
- potrafi wykorzystać **Python Console** w środowisku PyCharm,
- potrafi samodzielnie wywołać połączenie GET do zewnętrznego **API** w języku Python,
- dzieli się wiedzą z koleżankami i kolegami z zespołu,
- komunikuje się w sposób klarowny podczas dyskusji z kolegami i koleżankami w zespole.

METODY I FORMY PRACY:

- **oglądowe i słowne** - nauczyciel, po krótkim wprowadzeniu, odtworza i ewentualnie omawia filmy instruktażowe, zawierające prezentację czynności wykonywanych przez autora filmu,
- **praktyczne** - uczniowie indywidualnie lub w zespole podejmują się wykonania operacji zaprezentowanych w filmach, adaptując je do za-

stosowania w opracowywanym rozwiązaniu,

- **aktywizujące** - uczniowie, w ramach utworzonych zespołów, dyskutują nt. przyjęcia optymalnych rozwiązań w celu realizacji zaplanowanego projektu.

ŚRODKI DYDAKTYCZNE:

- Przed rozpoczęciem zajęć należy zweryfikować działanie VirtualBox na komputerach szkolnych - to konieczne w celu powodzenia zajęć.
- Przed zajęciami nauczyciel zapoznaje się z filmami, aby potrafił odpowiedzieć na pytania uczniów; w szczególności zalecamy filmy A20 i A21.



<https://diode.zone/w/p/u1rBNzA4XzSRMq8BrhurfW>

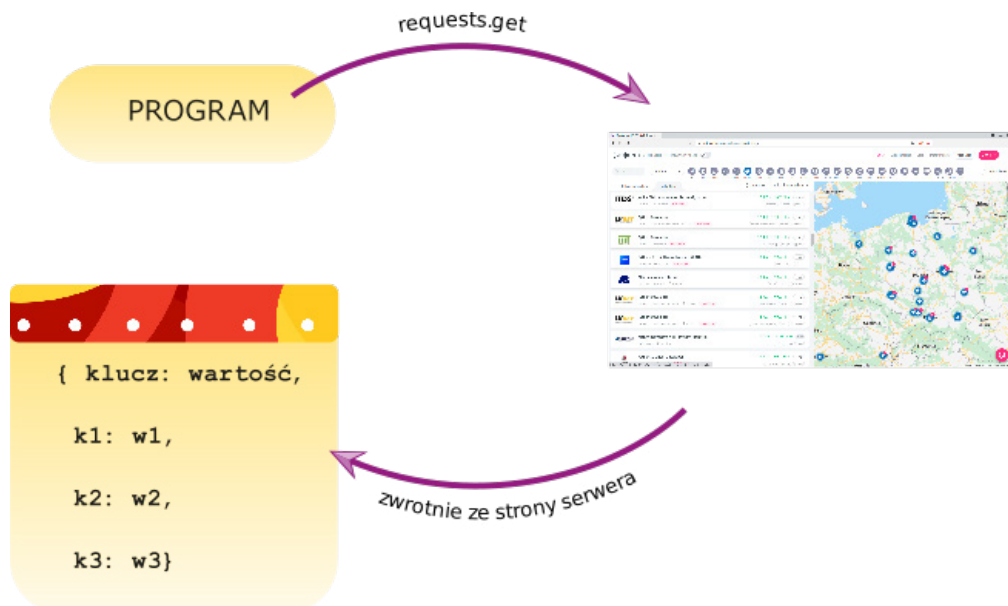
PRZEBIEG ZAJĘĆ:

Nauczyciel prezentuje filmy i komentuje je w tych momentach, które uzna za stosowne w zależności od wiedzy uczniów. Uczniowie oglądają, starają się wykonać te elementy, które potrafią, ewentualnie przy wsparciu nauczyciela wykonują je po każdym filmie. Poniżej lista filmów:

- Edytor tekstów: zrzut zawartości okna aplikacji i dodanie do tekstu (B14)
- Edytor grafiki: warstwy i dodanie tekstu. (B15)
- Edytor HTML: różne elementy na stronie (display, images, listy). (B16)
- Poznajemy Python Console w PyCharm + Wykorzystujemy requirements.txt i instalujemy niezbędne elementy: requests (A20)
- Wykonujemy requests.get() - z serwisu <https://fastapi.jurkiewicz.tech/> pokazujemy odczytane dane (A21)

WSKAZÓWKI METODYCZNE:

Przy tłumaczeniu działania `requests.get()` może być pomocna grafika .



- Podczas zajęć nauczyciel spotyka się z każdym zespołem osobno, weryfikuje, na jakim etapie znajduje się ich praca nad aplikacją i w razie potrzeby udziela wsparcia. Sprawdza, czy wszyscy członkowie zespołu są zaangażowani w wykonywanie zadań informatycznych.
- Zajęcia zostały zaplanowane do przeprowadzenia w czasie jednej jednostki lekcyjnej (45 minut). Jeśli jednak nauczyciel - ze względu na tempo pracy zespołów lub złożoność zagadnienia - uzna, że to za mało, zalecamy rozważenie takiej organizacji zajęć, aby można było dokończyć je na kolejnej jednostce lekcyjnej (dodatkowe do 45 minut).

MATERIAŁY POMOCNICZE:

Dokumentacja modułu requests: <https://requests.readthedocs.io/en/latest/>

ZAJĘCIA 7.

TEMAT: FORMAT JSON, PĘTLE, LISTY, SŁOWNIKI W PYTHON

CEL OGÓLNY ZAJĘĆ:

Wykorzystanie modułu `requests` do komunikacji z API. Analiza odpowiedzi w formacie JSON, z wykorzystaniem pętli iteracyjnej `for`.

Rozwijanie odpowiedzialności za pracę własną na rzecz wspólnego celu zespołu.

CELE SZCZEGÓŁOWE:

Uczeń:

- zna zasady tworzenia list i słowników w Python,
- potrafi wykorzystać pętlę iteracyjną `for` do przetworzenia obiektów sekwencyjnych,
- potrafi wykorzystać połączenie `requests.get()` w skrypcie,
- rozróżnia rodzaj `mutable` i `immutable` w Python,
- zachowuje się odpowiedzialnie w stosunku do pozostałych członków zespołu i wywiązuje się z własnych zobowiązań, mając na celu osiągnięcie wspólnego celu zespołu.

METODY I FORMY PRACY:

- **ogładowe i słowne** - nauczyciel, po krótkim wprowadzeniu, odtworza i ewentualnie omawia filmy instruktażowe, zawierające prezentację czynności wykonywanych przez autora filmu,
- **praktyczne** - uczniowie indywidualnie lub w zespole podejmują się wykonania operacji zaprezentowanych w filmach, adaptując je do zastosowania w opracowywanym rozwiązaniu,

- **aktywizujące** - uczniowie, w ramach utworzonych zespołów, dyskutują nt. przyjęcia optymalnych rozwiązań w celu realizacji zaplanowanego projektu.

ŚRODKI DYDAKTYCZNE

- Przed rozpoczęciem zajęć należy zweryfikować działanie VirtualBox na komputerach szkolnych - to konieczne w celu powodzenia zajęć.
- Przed zajęciami nauczyciel zapoznaje się z filmami, aby potrafił odpowiedzieć na pytania uczniów; w szczególności zalecamy filmy A24 i A25.



<https://diode.zone/w/p/2MRuJsB5Ps3AfmqeDXz24K>

PRZEBIEG ZAJĘĆ:

Nauczyciel prezentuje filmy i komentuje je w tych momentach, które uzna za stosowne w zależności od wiedzy uczniów. Uczniowie oglądają, starają się wykonać te elementy, które potrafią, ewentualnie przy wsparciu nauczyciela wykonują je po każdym filmie. Poniżej lista filmów:

- JSON i słowniki w Python (A22)
- Użycie pętli `for` dla pokazania elementów słownika z serwisu <https://fastapi.jurkiewicz.tech> (A23)
- Listy jako elementy słowników (A24)

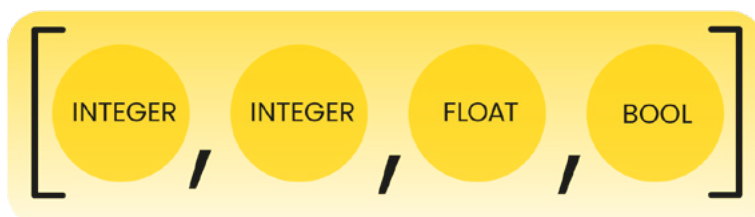


<https://tinyurl.com/prog-python-seq>

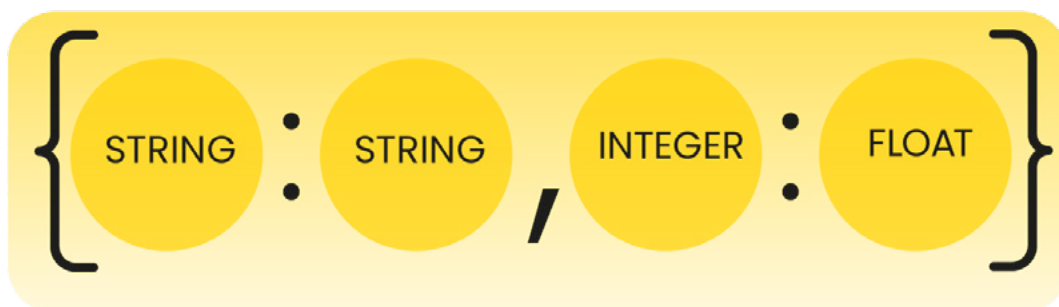
- Słowniki jako elementy słowników (A25)

WSKAZÓWKI METODYCZNE

- Warto zaznaczyć, że słownik może być zawsze elementem listy, a lista może być tylko wartością klucza słownika, nigdy samym kluczem!



Lista może zawierać dowolne typy jako kolejne elementy.



Słownik może zawierać tylko określone typy danych jako klucze, ale dowolne jako wartości.

Możemy posłużyć się przykładowym kodem, który pokaże błąd - zawarty jest w pliku `teachers_sample_excercises/07_dict_lists.py`; błąd wygląda następująco:

```
Traceback (most recent call last):
  File "teachers_sample_excercises/07_dict_lists.py", line 19, in <module>
    słownik = {"Linux", "Operating system": "System"}
TypeError: unhashable type: 'list'
```



- Zajęcia zostały zaplanowane do przeprowadzenia w czasie jednej jednostki lekcyjnej (45 minut). Jeśli jednak nauczyciel - ze względu na tempo pracy zespołów lub złożoność zagadnienia - uzna, że to za mało, zalecamy rozważenie takiej organizacji zajęć, aby można było dokończyć je na kolejnej jednostce lekcyjnej (dodatkowe do 45 minut).

MATERIAŁY POMOCNICZE

Rozdział 1 książki pt. „Python 3 - Projekty dla początkujących i pasjonatów”, wydawnictwo Helion Edukacja (str. 28 - sekwencyjne obiekty nieuporządkowane).

ZAJĘCIA 8.

TEMAT: EDYTORY ORAZ DOKUMENTACJA I API_KEY DLA PRZYKŁADOWYCH PROJEKTÓW

CEL OGÓLNY ZAJĘĆ:

Sprawdzenie dokumentacji oraz generowanie API_KEY różnych przykładowych API.

Rozwijanie odpowiedzialności za pracę własną na rzecz wspólnego celu zespołu.

CELE SZCZEGÓŁOWE:

Uczeń:

- zna różne formaty plików graficznych oraz tekstowych,
- potrafi wykorzystać różne obiekty w języku HTML,
- potrafi samodzielnie zalogować się do serwisu API i wygenerować tzw. API_KEY,
- zachowuje się odpowiedzialnie w stosunku do pozostałych członków zespołu i wywiązuje się z własnych zobowiązań, mając na celu osiągnięcie wspólnego celu zespołu.

METODY I FORMY PRACY:

- **ogładowe i słowne** - nauczyciel, po krótkim wprowadzeniu, odtworza i ewentualnie omawia filmy instruktażowe, zawierające prezentację czynności wykonywanych przez autora filmu,
- **praktyczne** - uczniowie indywidualnie lub w zespole podejmują się wykonania operacji zaprezentowanych w filmach, adaptując je do zastosowania w opracowywanym rozwiązaniu,
- **aktywizujące** - uczniowie, w ramach utworzonych zespołów, dys-

kutują nt. przyjęcia optymalnych rozwiązań w celu realizacji zaplanowanego projektu.

ŚRODKI DYDAKTYCZNE

- Przed rozpoczęciem zajęć należy zweryfikować działanie VirtualBox na komputerach szkolnych - to konieczne w celu powodzenia zajęć.
- Przed zajęciami nauczyciel zapoznaje się z filmami, aby potrafił odpowiedzieć na pytania uczniów; w szczególności zalecamy filmy A26, A27, A28.



<https://diode.zone/w/p/kgfVgxxKBN1uQ3cn98U8SL>

PRZEBIEG ZAJĘĆ:

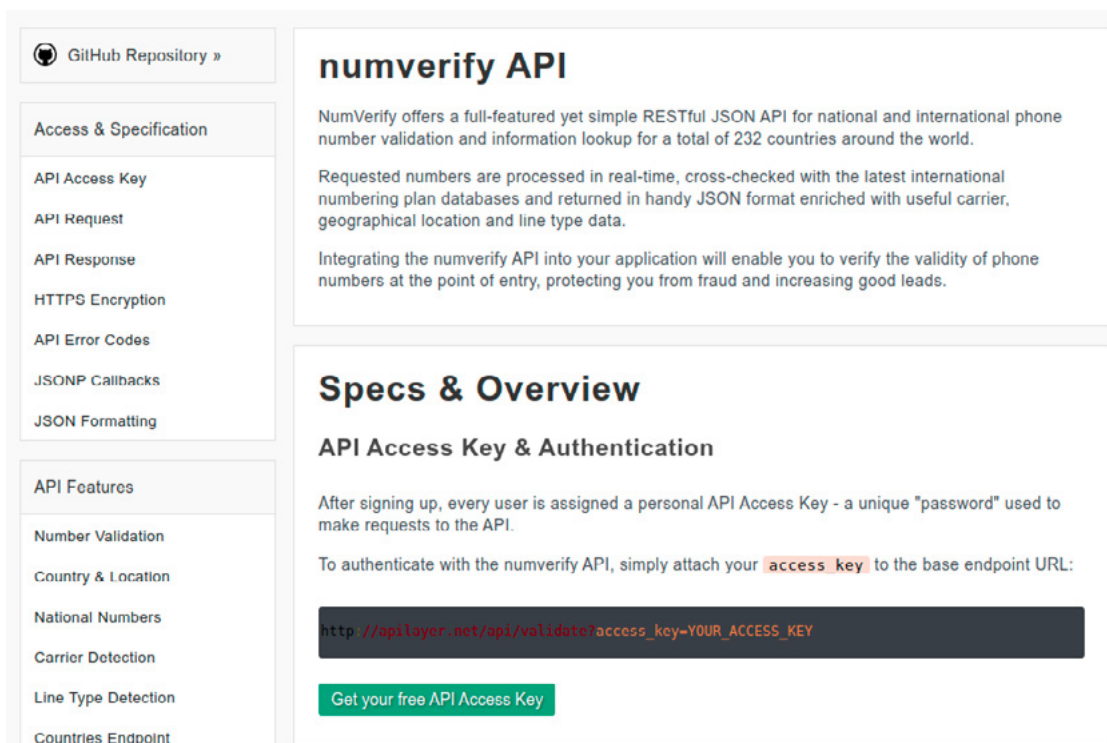
Nauczyciel prezentuje filmy i komentuje je w tych momentach, które uzna za stosowne w zależności od wiedzy uczniów. Uczniowie oglądają, starają się wykonać te elementy, które potrafią, ewentualnie przy wsparciu nauczyciela wykonują je po każdym filmie. Poniżej lista filmów:

- Edytor grafiki: eksport obrazu jako PNG (B17)
- Edytor tekstów: eksport dokumentu do formatu PDF (B18)
- Edytor HTML: różne elementy na stronie (navbar, listy, image) (B19)
- Sprawdzamy dokumentację dla przykładowych API: (A26, A27, A28)
 1. <https://aviationstack.com/documentation>
 2. <https://numverify.com/documentation>
 3. <https://wttr.in/:help>

- Generowanie API_KEY dla wybranego projektu (Aviationstack) - <https://aviationstack.com/signup/free> (A29)
- Generowanie API_KEY dla wybranego projektu (Numerify) - <https://numverify.com/documentation> (A30)
- Poznajemy kody odpowiedzi API: poprawnych i błędnych (A31, A32)

WSKAZÓWKI METODYCZNE

- Można wskazać uczniom dokumentację jako przykład szerokiego stosowania języka angielskiego - to może wspomóc zbudowanie motywacji do nauki języka obcego.



- Dla tych osób, które mają problemy, możemy zaproponować dowolny system tłumaczenia online, np. <https://translate.google.pl/>
- Zajęcia zostały zaplanowane do przeprowadzenia w czasie jednej jednostki lekcyjnej (45 minut). Jeśli jednak nauczyciel - ze względu na tempo pracy zespołów lub złożoność zagadnienia - uzna, że to za mało, zalecamy rozważenie takiej organizacji zajęć, aby można było dokończyć je na kolejnej jednostce lekcyjnej (dodatkowe do 45 minut).



UWAGA:

Warto zaznaczyć, aby uczniowie dokładnie zapoznali się z tymi filmami, zwłaszcza ci, którzy są odpowiedzialni za kwestie programistyczne. W tym momencie mogą już realnie tworzyć aplikację, ponieważ mają 75% informacji niezbędnych do jej wytworzenia. To spora dawka materiału, zwłaszcza warto zapoznać się z dokumentacjami projektu, który dana grupa chce realizować.

MATERIAŁY POMOCNICZE

Brak.

ZAJĘCIA 9.

TEMAT: FUNKCJE, ZASIĘG ZMIENNYCH - NAMESPACE, WŁASNE REPOZYTORIUM

CEL OGÓLNY ZAJĘĆ:

Utworzenie własnego repozytorium w serwisie GitHub oraz zapoznanie się ze sposobami definiowania funkcji w języku Python i przestrzeniami nazw (ang. namespaces).

Doskonalenie kompetencji współpracy w zespole.

CELE SZCZEGÓŁOWE:

Uczeń:

- zna podstawowe zasady funkcjonowania środowiska GitHub,
- potrafi stworzyć repozytorium, wybrać odpowiednią licencję dla oprogramowania,
- potrafi zdefiniować i wywołać funkcję w języku Python,
- wie, jak działają przestrzenie nazw i potrafi dostosować kod programu do odpowiednich przestrzeni,
- zachowuje się odpowiedzialnie w stosunku do pozostałych członków zespołu i wywiązuje się z własnych zobowiązań, mając na celu osiągnięcie wspólnego celu zespołu.

METODY I FORMY PRACY:

- **oglądowe i słowne** - nauczyciel, po krótkim wprowadzeniu, odtworza i ewentualnie omawia filmy instruktażowe, zawierające prezentację czynności wykonywanych przez autora filmu,
- **praktyczne** - uczniowie indywidualnie lub w zespole podejmują się

wykonania operacji zaprezentowanych w filmach, adaptując je do zastosowania w opracowywanym rozwiązaniu,

- **aktywizujące** - uczniowie, w ramach utworzonych zespołów, dyskutują nt. przyjęcia optymalnych rozwiązań w celu realizacji zaplanowanego projektu.

ŚRODKI DYDAKTYCZNE

- Przed rozpoczęciem zajęć należy zweryfikować działanie VirtualBox na komputerach szkolnych - to konieczne w celu powodzenia zajęć.
- Przed zajęciami nauczyciel zapoznaje się z filmami, aby potrafił odpowiedzieć na pytania uczniów; w szczególności zalecamy film A35.



<https://diode.zone/w/p/fYheNMwcS6w2Kyq6PXJMEw>

PRZEBIEG ZAJĘĆ:

Nauczyciel prezentuje filmy i komentuje je w tych momentach, które uzna za stosowne w zależności od wiedzy uczniów. Uczniowie oglądają, starają się wykonać te elementy, które potrafią, ewentualnie przy wsparciu nauczyciela wykonują je po każdym filmie. Poniżej lista filmów:

- Tworzymy własne repozytorium, pamiętamy o `.gitignore`, `README.md` oraz licencji (A33)
- Definiowanie funkcji w Python (A34)
- Funkcje i zasięg zmiennych w Python (A35)

WSKAZÓWKI METODYCZNE

- Dla tworzenia własnego repozytorium warto wskazać na możliwości wyboru różnych licencji na oprogramowanie:



<https://opensource.org/licenses>

- **Funkcja** - to zdefiniowany blok kodu, który realizuje pewne zadania, może bazować na różnych danych wejściowych, zwanych parametrami. Funkcja może „zwrócić” nam pewne wartości.

Ogólny sposób definiowania funkcji w Pythonie wygląda tak:

```
def nazwa_funkcji(parametr_1, parametr_2):  
    # blok kodu  
    return zwracany_obiekt
```

Zwróćmy uwagę na pewne ważne elementy:

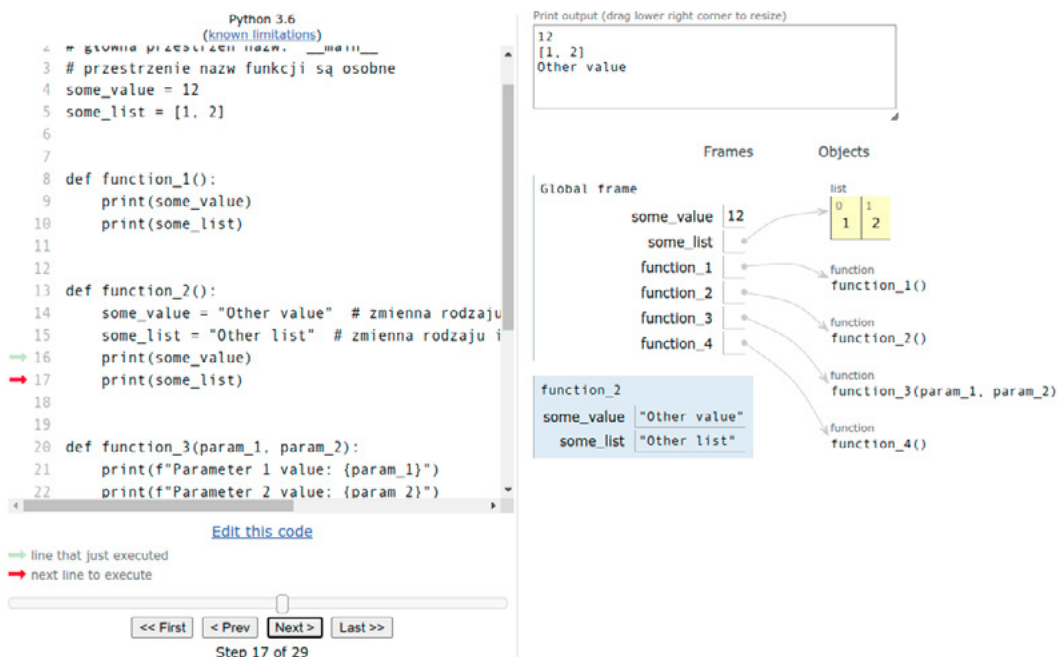
- słowo kluczowe `def` (ang. define) - mówi nam, że to będzie definicja funkcji,
- nazwa naszej funkcji zapisana jest małymi literami, jeśli zawiera dwa lub więcej słów oddzielamy je znakiem podkreślenia (to zasady z dokumentu PEP 8),
- parametry umieszczone są w nawiasie i rozdzielone przecinkami; wywołując funkcję musimy umieścić tyle samo argumentów, co parametrów w definicji (Uwaga! To nie jest do końca prawda, ale na tę chwilę przyjmujemy takie założenie, później, przy tłumaczeniu konkretnego kodu, poznamy więcej możliwości),
- dwukropek oznacza, że następnie występuje blok kodu i musi być zachowany,
- słowo kluczowe `return` (pol. zwróć/oddaj) - mówi nam, że to koniec działania funkcji i zwraca `zwracany_obiekt`
- Zajęcia zostały zaplanowane do przeprowadzenia w czasie jednej jednostki lekcyjnej (45 minut). Jeśli jednak nauczyciel - ze względu na

tempo pracy zespołów lub złożoność zagadnienia - uzna, że to za mało, zalecamy rozważenie takiej organizacji zajęć, aby można było dokończyć je na kolejnej jednostce lekcyjnej (dodatkowe do 45 minut).

MATERIAŁY POMOCNICZE

Przestrzenie nazw (ang. namespace) to skomplikowany temat - można wykorzystać wizualizację pod adresem: <https://tinyurl.com/popo-namespac>

Python Tutor: Visualize code in Python, JavaScript, C, C++, and Java



The screenshot shows the Python Tutor interface. On the left, a code editor displays Python 3.6 code with comments in Polish. The code defines a global namespace and several functions. The execution is paused at line 17. On the right, the 'Frames' and 'Objects' panels are visible. The 'Global frame' contains variables 'some_value' (12) and 'some_list' ([1, 2]). The 'function_2' frame shows 'some_value' as 'Other value' and 'some_list' as 'Other list'. The 'Objects' panel shows the memory layout of these variables and functions.

Dodatkowy materiał o przestrzeniach nazw:



<https://python.szkola.pl/course/koncepcja-przestrzeni-nazw-w-pythonie-i-widocznosc-zmiennych/>

ZAJĘCIA 10.

TEMAT: TESTY DOSTĘPU DO API ORAZ WSPÓŁPRACA Z GITHUB'EM

CEL OGÓLNY ZAJĘĆ:

Testy dostępu do systemów API. Replikowanie projektu z serwisu GitHub na lokalny dysk. Umieszczenie aplikacji w repozytorium.

Doskonalenie kompetencji współpracy w zespole.

CELE SZCZEGÓŁOWE:

Uczeń:

- zna podstawowe zasady funkcjonowania środowiska GitHub,
- potrafi zmienić ustawienia konta w serwisie GitHub (dodać współpracowników),
- potrafi samodzielnie skonfigurować wirtualne środowisko dla pojedynczego lokalnego projektu w oparciu o przedstawiony schemat,
- testuje dostęp do zewnętrznego **API** za pomocą skryptu,
- zachowuje się odpowiedzialnie w stosunku do pozostałych członków zespołu i wywiązuje się z własnych zobowiązań, mając na celu osiągnięcie wspólnego celu zespołu.

METODY I FORMY PRACY:

- **oglądowe i słowne** - nauczyciel, po krótkim wprowadzeniu, odtworza i ewentualnie omawia filmy instruktażowe, zawierające prezentację czynności wykonywanych przez autora filmu,
- **praktyczne** - uczniowie indywidualnie lub w zespole podejmują się wykonania operacji zaprezentowanych w filmach, adaptując je do zastosowania w opracowywanym rozwiązaniu,

aktywizujące - uczniowie, w ramach utworzonych zespołów, dyskutują nt. przyjęcia optymalnych rozwiązań w celu realizacji zaplanowanego projektu.

ŚRODKI DYDAKTYCZNE:

- Przed rozpoczęciem zajęć należy zweryfikować działanie VirtualBox na komputerach szkolnych - to konieczne w celu powodzenia zajęć.
- Przed zajęciami nauczyciel zapoznaje się z filmami, aby potrafił odpowiedzieć na pytania uczniów; w szczególności zalecamy filmy A39 i A40 - aby przetestować proces replikacji i dodawanie lokalnego venv.



<https://diode.zone/w/p/7DeNCrqLzhHw6i72tjxmmK>

PRZEBIEG ZAJĘĆ:

Nauczyciel prezentuje filmy i komentuje je w tych momentach, które uzna za stosowne w zależności od wiedzy uczniów. Uczniowie oglądają, starają się wykonać te elementy, które potrafią, ewentualnie przy wsparciu nauczyciela wykonują je po każdym filmie.

Należy zwrócić uwagę na zmienną `access_key` (to odpowiednik `API_KEY`), która zawiera przykładowy klucz dostępu - na 99,999% będzie on nieaktywny w momencie testowania, więc uczniowie muszą użyć swoich kluczy, które wygenerowane zostały na wcześniejszych zajęciach.


```
# test naszego API
import requests
import sys

params = {
    'access_key': "4f0cb9a05ca9e4b845e7fbd8002ac32d",
    'flight_status': 'active',
}

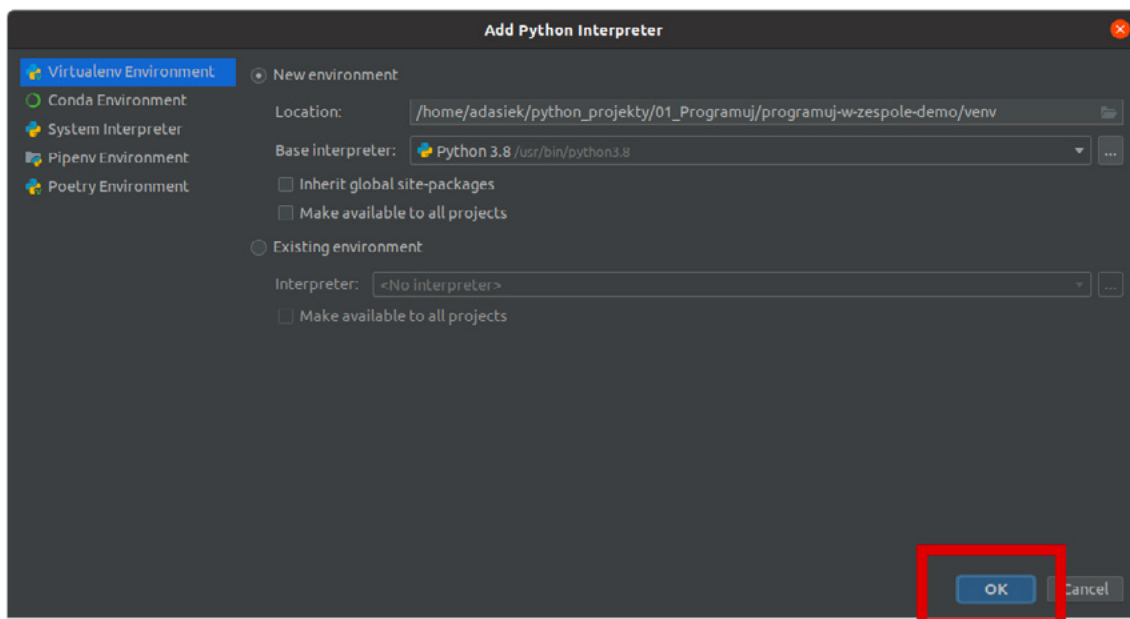
api_result = requests.get('http://api.aviationstack.com/v1/flights', params)
```

Poniżej lista filmów:

- Testujemy dostęp do danych API (aviationstack) (A36)
- Testujemy dostęp do danych API (numverify) (A37)
- Testujemy dostęp do danych API (wttr.in) (A38)
- Replikacja projektu z GitHub do PyCharm (open via VCS) i dodanie lokalnego venv (Add interpreter) i w requirements.txt Install all packages (A39)
- Dodanie do repozytorium pracy z aplikacją PySimpleGUI i Commit/Push (A40)

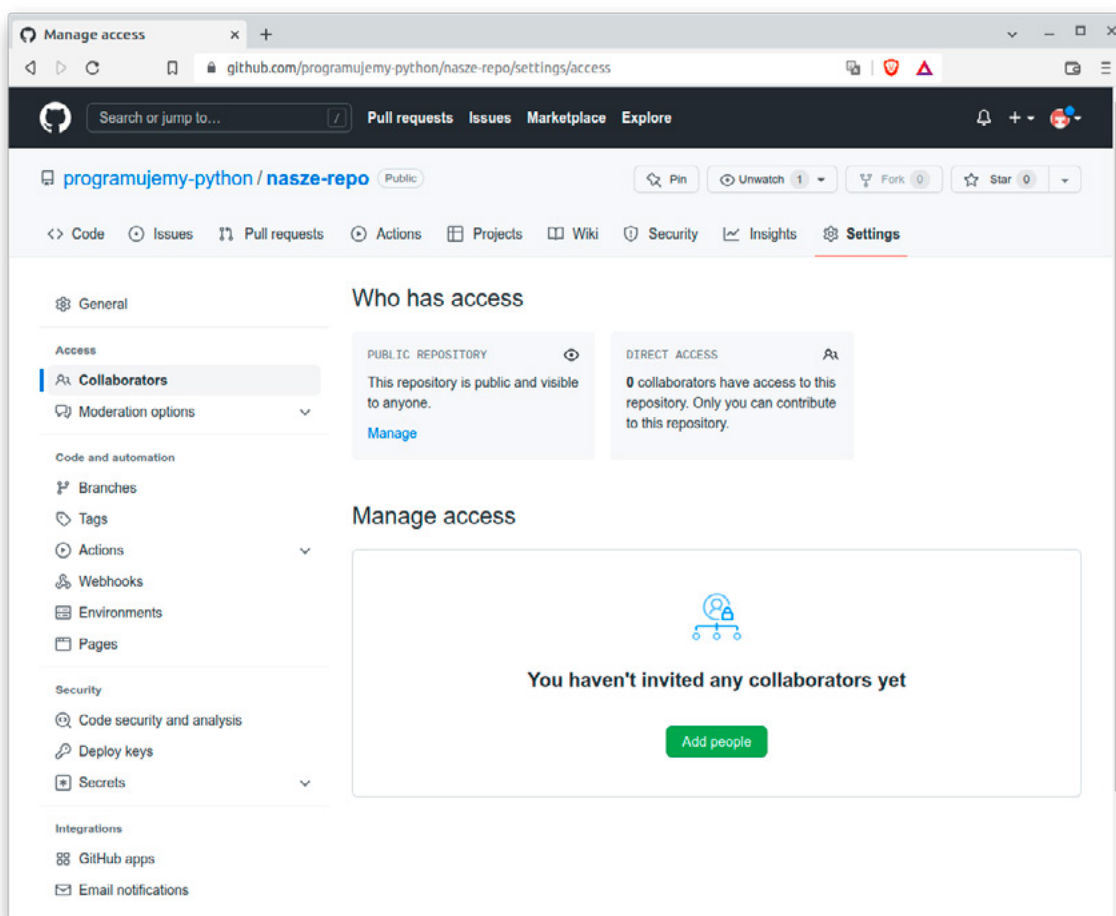
WSKAZÓWKI METODYCZNE:

Pamiętajmy o tym, jak aktywować w PyCharm Interpreter Python dla venv:



Ważne

- Na koniec lekcji liderzy zespołów dodają aplikację do repozytorium projektu w serwisie GitHub i udzielają innym do niego dostępu.



Dzięki temu pozostałe osoby w zespole mogą dodać do repozytorium własne pliki, również replikując projekt do PyCharm, dodają (kopiując) pliki i następnie wykonują Commit/Push.

- Zajęcia zostały zaplanowane do przeprowadzenia w czasie jednej jednostki lekcyjnej (45 minut). Jeśli jednak nauczyciel - ze względu na tempo pracy zespołów lub złożoność zagadnienia - uzna, że to za mało, zalecamy rozważenie takiej organizacji zajęć, aby można było dokończyć je na kolejnej jednostce lekcyjnej (dodatkowe do 45 minut).

MATERIAŁY POMOCNICZE:

- Oficjalna dokumentacja techniczna dla języka Python 3 (lub nowszej wersji), dostępna pod adresem: <https://docs.python.org/>
- Dokumentacja PyCharm



https://www.jetbrains.com/help/pycharm/creating-virtual-environment.html#python_create_virtual_env

Ciekawe pytanie w serwisie Stackoverflow dotyczące Virtual env w PyCharm:



<https://stackoverflow.com/questions/23262837/how-do-setup-virtualenv-and-dependencies-in-pycharm>

ZAJĘCIA 11.

TEMAT: SPRAWDZANIE POŁĄCZEŃ DO API , MANIPULOWANIE ELEMENTAMI SŁOWNIKÓW

CEL OGÓLNY ZAJĘĆ:

Komunikacja komputera z systemami API, testy oraz aktualizacja repozytorium w serwisie GitHub. Manipulowanie elementami słowników.

Doskonalenie kompetencji współpracy w zespole.

CELE SZCZEGÓŁOWE:

Uczeń:

- zna podstawowe zasady funkcjonowania środowiska GitHub,
- potrafi analizować kod programu w poszukiwaniu błędów,
- testuje dostęp do zewnętrznego **API** za pomocą skryptu,
- zachowuje się odpowiedzialnie w stosunku do pozostałych członków zespołu i wywiązuje się z własnych zobowiązań, mając na celu osiągnięcie wspólnego celu zespołu.

METODY I FORMY PRACY:

- **oglądowe i słowne** - nauczyciel, po krótkim wprowadzeniu, odtworza i ewentualnie omawia filmy instruktażowe, zawierające prezentację czynności wykonywanych przez autora filmu,
- **praktyczne** - uczniowie indywidualnie lub w zespole podejmują się wykonania operacji zaprezentowanych w filmach, adaptując je do zastosowania w opracowywanym rozwiązaniu,
- **aktywizujące** - uczniowie, w ramach utworzonych zespołów, dyskutują nt. przyjęcia optymalnych rozwiązań w celu realizacji zaplanowanego projektu.

ŚRODKI DYDAKTYCZNE:

- Przed rozpoczęciem zajęć należy zweryfikować działanie VirtualBox na komputerach szkolnych - to konieczne w celu powodzenia zajęć.
- Przed zajęciami nauczyciel zapoznaje się z filmami, aby potrafił odpowiedzieć na pytania uczniów.



<https://diode.zone/w/p/51UYhfeH2NyZGwRL98vCWM>

- Zalecamy analizę skryptu: `teachers_sample_excercises/11_dicts.py`, aby przygotować się do przedstawienia błędu i ewentualnego zadania pracy domowej.

PRZEBIEG ZAJĘĆ:

Nauczyciel w trakcie prezentacji filmów komentuje je w tych momentach, które uzna za stosowne w zależności od wiedzy uczniów. Uczniowie oglądają i starają się wykonać te elementy, które potrafią, ewentualnie przy wsparciu nauczyciela wykonują je po każdym filmie. Poniżej lista filmów:

- Sprawdzenie działania — skrypt odczytujący i prezentujący wybrane dane (A41)
- Wysyłanie projektu do serwisu GitHub (A42)
- Aktualizowanie wartości dla kluczy słowników (A43)
- Tworzenie słowników i dodawania do nich elementów (A44)

Warto zaznaczyć, że to ostatnia dawka teorii - w tym momencie ok. 80% projektu powinno już być wykonane.

- Tworzenie pełnej aplikacji



- Tworzenie docelowego interfejsu aplikacji
- Przygotowanie końcowe dokumentacji i strony w HTML

WSKAZÓWKI METODYCZNE:

- Warto tu zwrócić uwagę na możliwe błędy podczas odwołania się do nieistniejącego klucza, jeśli chcemy zobaczyć wartość - kod zawarty w pliku: `teachers_sample_excercises/11_dicts.py`:

```
days = {
    0: {"min_temp": 0, "max_temp": 0, "warning": False},
    1: {"min_temp": 0, "max_temp": 0, "warning": False},
    2: {"min_temp": 0, "max_temp": 0, "warning": False},
}
print(days[1]["min_temp"])
print(days[2]["max-temp"]) # tu wystąpi błąd
# Traceback (most recent call last):
#   File "teachers_sample_excercises/11_dicts.py", line 7, in <module>
#     print(days[2]["max-temp"]) # tu wystąpi błąd
# KeyError: 'max-temp'
```

Możemy zadać jako zadanie domowe napisanie tego kodu w taki sposób, aby wyeliminować ewentualność błędu wykorzystując tylko instrukcję warunkową `if`, a nie konstrukcję `try ... except ...`. Przykładowe rozwiązanie zawarte jest w pliku.

- Zajęcia zostały zaplanowane do przeprowadzenia w czasie jednej jednostki lekcyjnej (45 minut). Jeśli jednak nauczyciel - ze względu na tempo pracy zespołów lub złożoność zagadnienia - uzna, że to za mało, zalecamy rozważenie takiej organizacji zajęć, aby można było dokończyć je na kolejnej jednostce lekcyjnej (dodatkowe do 45 minut).



MATERIAŁY POMOCNICZE:

Wpis w serwisie Stackoferflow:



<https://stackoverflow.com/questions/28701878/dicts-in-python>



ZAJĘCIA 12.

TEMAT: PREZENTACJA REZULTATÓW PRACY ZESPOŁÓW I PODSUMOWANIE PROJEKTU

CEL OGÓLNY ZAJĘĆ:

Podsumowanie cyklu zajęć i udzielenie informacji zwrotnej nt. efektów pracy zespołów.

CELE SZCZEGÓŁOWE:

Uczeń:

- ocenia własne kompetencje współpracy i komunikacji w zespole,
- porównuje efekty pracy swojego zespołu i pozostałych zespołów,
- uzasadnia przyjęte sposoby rozwiązywania problemów, zarówno programistycznych, jak i interpersonalnych w ramach zespołu,
- obiektywnie wypowiada się na temat prezentacji innych zespołów.

METODY I FORMY PRACY:

- **ogładowe i słowne** - nauczyciel ocenia i komentuje wykonane aplikacje i produkty końcowe pracy zespołów, podsumowuje efekty zrealizowanego cyklu zajęć,
- **praktyczne** - uczniowie prezentują wyniki pracy swoich zespołów, uzasadniają i wyjaśniają decyzje podejmowane w trakcie wykonywania zadania, dokonują samodzielnie końcowej oceny kompetencji komunikacji i współpracy w zespole,
- **aktywizujące** - uczniowie komentują wyniki prac innych zespołów i udzielają sobie wzajemnie informacji zwrotnej.

ŚRODKI DYDAKTYCZNE:

- Nauczyciel dokonuje oceny końcowej prac poszczególnych zespołów – ocenia, czy zespół zrealizował swój projekt, tj. osiągnął zamierzony cel, czy przygotowana prosta aplikacja działa poprawnie (zgodnie z intencją zespołu), a kod jest poprawny i dobrej jakości. (Oceny stopnia osiągnięcia celów nie należy mylić z ocenami szkolnymi),
- Nauczyciel, w oparciu o wcześniejszą decyzję podjętą przed rozpoczęciem zajęć 0 i zakomunikowaną uczniom na starcie projektu, decyduje o przyznaniu nagrody zespołom (w zależności od rodzaju nagrody, zostanie ona przyznana jednemu, kilku lub wszystkim składom uczniowskim),
- Nauczyciel zapewnia możliwość prezentacji prac uczniów np. przy użyciu projektora, komputera z zainstalowanym środowiskiem PyCharm etc. ,
- Nauczyciel przygotowuje *Test umiejętności współpracy i komunikacji w zespole*.

PRZEBIEG ZAJĘĆ:

Nauczyciel zaprasza uczniów do prezentacji prac zespołowych.

Uczniowie biorą udział w dyskusji na temat prezentacji.

Nauczyciel moderuje dyskusję.

Podsumowuje zajęcia zarówno w obszarze informatycznym jak i pracy zespołowej.

Przeprowadza *Test umiejętności współpracy i komunikacji w zespole* (ok. 15 minut) i archiwizuje wyniki.

Dokonuje analizy zmiany poziomu samooceny kompetencji komunikacji i współpracy w zespole, porównując wyniki testu wstępnego i końcowego. Komentuje wyniki w obecności uczniów (podając pewne uogólnienia i statystyki, a nie wyniki poszczególnych uczniów).

Podsumowuje zajęcia "Progr@muji w zespole", dziękuje uczniom za zaangażowanie i wykonaną pracę. Jeśli to możliwe, proponuje uczniom jakąś formę wyróżnienia, np. możliwość upublicznienia (za ich zgodą) wyników prac na forum szkoły (w postaci aplikacji i stron www).

WSKAZÓWKI METODYCZNE:

- W trakcie zajęć uczniowie otrzymają możliwość prezentacji efektów pracy zespołów, w których pracowali nad rozwiązaniem/aplikacją. Nauczyciel powinien postarać się o taką organizację zajęć, aby każdy zespół otrzymał szansę na zaprezentowanie aplikacji. Jeśli zaprezentowanie i omówienie wyników pracy zespołów oraz wypełnienie testów kompetencji nie będzie możliwe w trakcie jednej jednostki lekcyjnej (45 minut), zalecamy rozważenie takiej organizacji zajęć, aby można było dokończyć je na kolejnej jednostce lekcyjnej (dodatkowe do 45 minut).
- Przy komentowaniu i sprawdzaniu prac można pomocniczo posłużyć się odpowiednim rozwiązaniem IT, np. formatterem kodu Black (<https://black.readthedocs.io/en/stable/>) i sprawdzić, ile zmian wykona. Warto też pokazać uczniom to narzędzie (lub inne wykorzystane w tym celu) i powiedzieć kilka słów nt. jego funkcjonalności i zastosowania.

MATERIAŁY POMOCNICZE:

Artykuł zawierający porady dot. przygotowania i wygłoszenia krótkich (5-minutowych) prezentacji:

<https://ahaslides.com/pl/blog/how-to-make-a-5-minute-presentation/>

Artykuł anglojęzyczny nt. dostępnych narzędzi do dokonywania przeglądu kodu programistycznego w języku Python:

<https://dzone.com/articles/7-best-python-code-review-tools-recommended-by-dev>