

Uniwersytet Mikołaja Kopernika w Toruniu
Wydział Matematyki i Informatyki

Alicja Kluczek
nr albumu: 324587
kierunek: informatyka

Praca magisterska

Nauka programowania dla klas 7-8 zintegrowana z platformą Szkopuł.

Opiekun pracy dyplomowej
prof. dr hab. Anna Beata
Kwiatkowska

Toruń 2025

Spis treści

Wstęp	5
1. Z czego składa się kurs? Filary programu.	7
Bibliografia	13

Wstęp

Współczesna edukacja informatyczna w szkołach podstawowych powinna rozwijać praktyczne umiejętności uczniów i przygotowywać ich do świadomego korzystania z technologii. Celem niniejszej pracy magisterskiej jest stworzenie kursu programowania dla klas 7–8, zintegrowanego z platformą Szkopuł, obejmującego 24 jednostek lekcyjnych, zestawy zadań i quizów, a także materiały ewaluacyjne – w tym klasówki i projekty zespołowe. Kurs jest zgodny z podstawą programową i może być wykorzystywany w ramach nauczania w szkole.

Oprócz części dydaktycznej praca obejmuje również rozwój funkcjonalności platformy – wdrożenie modułu Omówienia, który umożliwia dodawanie i kontrolę dostępu do komentarzy do zadań. Całość została zaprojektowana z myślą o praktycznym podejściu do nauczania informatyki i może zostać rozszerzona o elementy sztucznej inteligencji, np. w projektach semestralnych.

Rezultatem pracy będzie:

- Działający kurs opublikowany na Szkopuł Kursy,
- Komplet konspektów i materiałów do każdej opisanej lekcji,
- Nowy komponent „Omówienia” platformy Szkopuł.

Rozdział 1.

Z czego składa się kurs? Filary programu.

Program rozpoczyna się od nauki podstaw programowania, tak aby uczniowie mogli rozwijać umiejętność myślenia komputacyjnego na dalszych etapach kursu. Każda jednostka dydaktyczna składa się z części teoretycznej (w tym elementów „unplugged”) oraz zadań praktycznych realizowanych przy użyciu języka Python.

Ćwiczenia informatyki „unplugged” są według nowoczesnych badań co najmniej tak efektywne, jak standardowe metody nauki [?].

Plan zajęć obejmuje 24 godziny lekcyjne; podana liczba godzin ma charakter orientacyjny i może być dostosowywana w zależności od potrzeb grupy.

Myślenie algorytmiczne, podstawy programowania (12 godzin)

Moduł ten koncentruje się na wprowadzeniu uczniów w abstrakcyjne pojęcia programistyczne, takie jak zmienne, instrukcje warunkowe, pętle oraz funkcje. Zajęcia te przeplatane są mini-projektami, które umożliwiają uczniom praktyczne zastosowanie zdobytej wiedzy oraz wyrażenie własnej kreatywności.

Takie podejście zwiększa szansę na zaangażowanie uczniów, którzy niekoniecznie wykazują naturalne predyspozycje matematyczne. Abstrakcyjne koncepty nadal odgrywają istotną rolę, ponieważ rozwijają zdolność adaptacji do dynamicznie zmieniających się technologii informatycznych [?].

Badania wskazują również, że dziewczęta częściej interesują się praktycznym zastosowaniem technologii niż jej konstrukcją techniczną [?]. Włączenie projektowych i aplikacyjnych elementów do zajęć może więc pozytywnie wpływać na ich aktywność i zaangażowanie w naukę informatyki.

1. Podstawy Pythona – zmienne, operatory, `print()`, `input()`. Po co programować?

2. Myślenie komputacyjne od kuchni – specyfikacja, algorytm. Co to znaczy rozwiązać problem?
3. Instrukcje warunkowe – `if`, `elif`, `else`. Jak kierować zachowaniem komputera?
4. Pętle `for` i `while` – iteracje, przerwania (`break`, `continue`). Jak sobie poradzić z powtarzalną pracą? **[podwójna godzina]**
5. Zastosowanie praktyczne: Quiz. Jak użyć tego co umiem? **[podwójna godzina]**
6. Funkcje w Pythonie – argumenty, wartości zwracane, funkcje wbudowane.
7. Matematyka z pomocą Pythona – Jak napisać program rozwiązuający moją pracę domową?
8. Lista, słownik – Kiedy zmienna nie wystarcza.
9. Zastosowanie praktyczne – Sterowanie „robotem”. Biblioteka `turtle`. **[podwójna godzina]**

Kryptografia (4 godziny)

Choć programowanie stanowi kluczowy element wprowadzający do informatyki, nie wyczerpuje całego spektrum tej dziedziny. Włączenie krótkich modułów tematycznych, takich jak kryptografia klasyczna, pozwala uczniom lepiej zrozumieć szerokość informatyki jako nauki. Zgodnie z podejściem stosowanym w szkołach średnich w Szwajcarii [?], kryptografia może być skutecznym sposobem rozwijania umiejętności analitycznego myślenia oraz krytycznego podejścia do informacji – nawet bez wcześniejszej wiedzy z zakresu informatyki. Szczególną rolę odgrywa tu motywacja uczniów: element rywalizacji między twórcami a łamaczami kodów stanowi silny czynnik angażujący. W badaniach nad tego typu modułami wykazano, że aż 70% uczniów wykazało chęć dalszego zgłębiania tematu, co czyni z kryptografii znakomity punkt wyjścia do dalszych zajęć.

1. Podstawy kryptografii – Poznajemy szyfr Cezara, `rot13`. Dlaczego są słabym za-abezpieczeniem?
2. Kody ASCII – zmiana liter w liczbę.
3. Implementacja wybranej metody szyfrowania – podstawieniowej, przesunięć, *leet speak*, Rozier. **[podwójna godzina]**

Algorytmika z Pythonem (8 godzin)

W tej części kursu uczniowie zapoznają się z podstawowymi algorytmami zawartymi w polskiej podstawie programowej dla klas 7–8. Analiza algorytmów odbywa się zarówno na poziomie koncepcyjnym (na kartce), jak i implementacyjnym (w języku Python).

Ponieważ algorytmika jest dziedziną o wysokim stopniu abstrakcji, szczególną uwagę poświęca się wizualizacji działania omawianych algorytmów. W procesie nauczania wykorzystywane są elementy gier, zabaw oraz komputerowe narzędzia wizualizacyjne, które są kluczowe by uczniowie mogli dobrze zrozumieć mechanizmy stojące za poszczególnymi rozwiązaniami [?].

1. Własności liczbowe – podzielność, suma cyfr, iloczyn cyfr. **[podwójna godzina]**
2. Kod binarny, szesnastkowy.
3. Algorytm Euklidesa – zastosowanie.
4. Wyszukiwanie w zbiorze – Liniowo znajdujemy minimum, maksimum, k-ty element. **[podwójna godzina]**
5. Porządkowanie zbioru – sortowanie przez wybieranie. **[podwójna godzina]**

Bibliografia

- [1] On plugging "unplugged" into CS classes, 2013
- [2] Are Females Disinclined to Tinker in Computer Science? 2015
- [3] L'enseignement de l'Informatique en France: Il est urgent de ne plus attendre, 2013
- [4] A short introduction to classical cryptology as a way to motivate high school students for informatics, 2011
- [5] Exploring the role of visualization and engagement in computer science education