

Specjalizacja Techniczno-Informatyczna

Na specjalizacji techniczno-informatycznej uczymy:

- przewidywania zagrożeń i zapobiegania im, korzystając z odpowiednich środków ochrony osobistej,
- projektowania i wykonywania prototypów i urządzeń,
- czytania i rysowania prostych schematów elektrycznych i rysunków technicznych,
- podstaw metrologii, czyli wykonywania dokładnych pomiarów wymiarów i cech przedmiotów,
- obsługi obrabiarek (drukarek 3d, tokarki, wycinarki laserowej, przecinarki plazmowej, przecinarki taśmowej),
- obsługi elektronarzędzi (szlifierki kątowej, wiertarki, piły ukosowej, pilarki, wyrzynarki, frezarki, wkrętarki, zakrętarki),
- spawania metodami TIG, MIG/MAG i MMA,
- podstaw obsługi robota przemysłowego KUKA,
- programowania w językach C++, Python i C#,
- budowania aplikacji okienkowych w Visual Studio w języku C#,
- programowania mikrokontrolerów AVR na platformie Arduino,
- podstaw elektroniki analogowej i cyfrowej,
- projektowania, budowy i programowania częściowo autonomicznych robotów mobilnych,
- umiejętności grupowej pracy projektowej łączącej elementy mechaniki, elektroniki i programowania

Umożliwiamy również startowanie w różnorodnych konkursach elektronicznych/robotycznych/programistycznych. Do użytku osób uczniowskich pozostaje pracownia techniczna, warsztat w piwnicy i siedziba zaprzyjaźnionego stowarzyszenia Warszawski Hackerspace. W miejscach tych os. uczniowskie mogą realizować projekty własne, ze wsparciem merytorycznym nauczycieli specjalizacji.

W przypadku zdobycia dodatkowych środków:

- osoby uczniowskie będą mogły zabierać swoje dzieła ze sobą (obecnie często musimy rozbierać roboty czy konstrukcje by odzyskać elementy mechaniczne, elektroniczne, mikrokontrolery itp.),
- będziemy mogli zakupić różne materiały - drewno metal plexi - do wykorzystania przez os. uczniowskie w ich projektach,

będziemy mogli doposażyć pracownię i warsztat w:

- drukarkę żywiczną (możliwość wytwarzania skomplikowanych obiektów, o dużej wytrzymałości i wysokiej dokładności),
- ploter laserowy (do szybkiego prototypowania w sklejkę, materiałach drewnopochodnych i plexi),
- małą tokarkę (do przedstawienia procesu toczenia i wykonywania prostych, małych metalowych elementów na miejscu w szkole),
- małą frezarkę np. proxxon (jak w tokarce do procesu i wykonywania małych detali),
- oscyloskop cyfrowy (dokładne badanie szybkich przebiegów, dekodowanie wybranych protokołów).

Szczegółowy opis specjalizacji

Łącznie nauczycieli na specjalizacji: 4 osoby + 1 os. pre-specjalizacyjnie

Klasa I

Adam Mirowski - obowiązkowy - 2h (Cyfrowe zajęcia praktyczno-techniczne)

Monika Bucewka - obowiązkowy - 2h (Informatyka)

Klasa II

Adam Mirowski - obieralny - 4h (Informatyka programistyczna)

Jerzy Piwkowski - obieralny - 4h (Informatyka programistyczna)

Adam Mirowski - obieralny - 2h (Zajawka specjalizacji)

Klasa III

Adam Mirowski - obowiązkowy - 2h (Projekt)

Jerzy Piwkowski - obowiązkowy - 2h (Podstawy projektowania i wytwarzania)

Paweł Kotowski - obowiązkowy - 2h (Programowanie zaawansowane I)

Filip Krynicki - obowiązkowy - 2h (Repetytorium I)

Klasa IV

Adam Mirowski - obowiązkowy - 4h (Hackerspace)

Paweł Kotowski - obowiązkowy - 2h (Programowanie zaawansowane II)

Filip Krynicki - obowiązkowy - 2h (Repetytorium II)

Szczegółowy opis przedmiotów

Klasa I

CZPT - Cyfrowe zajęcia praktyczno-techniczne -

- Domowe naprawianie
 - klejenie
 - skręcanie
 - zbijanie
 - malowanie
 - szpachlowanie
 - szlifowanie
 - szycie
 - wyłącznik nadprądowy i bezpiecznik topikowy
- Elektronika
 - czytanie i tworzenie schematów elektrycznych
 - symbole podstawowych komponentów elektronicznych
 - obsługa multimetru (voltomierz, amperomierz, omomierz)
 - łączenie szeregowo, równoległe
 - rodzaje czujników, zasada działania i zastosowanie
 - elementy wykonawcze
 - mikrokontrolery
- Elektryka
 - metody generowania prądu
 - podstawowe wiadomości o prądzie wchodzące w zakres podstawy programowej z fizyki: pojęcie napięcia, natężenia, oporu, pracy i mocy prądu, prawo Ohma, prawa Kirchhoffa
 - źródło zasilania, sposoby łączenia ogniw
- Informatyka

- podstawowe komponenty komputera (GPU, CPU, RAM, harddrive, motherboard, karty rozszerzeń, zasilacz)
- wymiana pasty termoprzewodzącej
- regularne czyszczenie wnętrza komputera z kurzu
- bezpieczne i dokładne czyszczenie wiatraków i radiatorów
- Mechanika
 - używanie podstawowych narzędzi - śrubokręta, kombinerek, wiertarki
 - podstawowe techniki montażu - taśma, kleje, śruby, gwoździe
 - rodzaje stosowanych napędów i przekładni
 - silniki elektryczne - zasada działania, rodzaje i zastosowania
 - proste mechanizmy hydrauliczne - tłoki ze strzykawek
- Projektowanie i drukowanie 3D
 - metody i środowiska modelowania 3D
 - materiały
 - metody drukowania 3D
- Programowanie (C++)
 - podstawowe operacje, ale tłumaczone w sposób mniej algorytmiczny/szkolny/studencki, a bardziej wyjście od rzeczywistości
 - instrukcja warunkowa
 - pętla
 - zmienne
 - funkcje

Inf- Informatyka -

- Programowanie (Python)
 - algorytmika
 - podstawowe kolekcje zmiennych
 - instrukcja warunkowa
 - iteracja
 - pętla
 - funkcje
 - algorytmy rekurencyjne
- Word
 - metody układania tekstu
 - tabulacja
 - prawidłowe formatowanie tekstu
- Excel
 - podstawowe funkcje
 - wykresy
 - tabele przestawne
 - sortowanie

Klasa II

Inf2- Informatyka programistyczna -

- Wstęp do informatyki programistycznej
 - omówienie czym jest komputer
 - podróż od tranzystorowej bramki logicznej po nowoczesny komputer
- Linux
 - podstawowe informacje o systemie i jego zastosowaniu
 - korzystanie z terminala
 - obsługa plików - podstawowe komendy
 - tworzenie dokumentów tekstowych (VIM i nano)

- o proste skrypty w bashu
- Programowanie (C++ i Python)
 - o powtórzenie wiadomości z klasy pierwszej: zmienne, instrukcja warunkowa, pętle, funkcje
 - o projekt gry snake w terminalu
 - o paradygmat programowania obiektowego
- Zagadnienia kierunkowe
 - o Aplikacje na telefon
 - o Robotyka mobilna
 - o Widzenie maszynowe
 - o WebDev
 - o GameDev
 - o Sieci neuronowe

Proj - Zajęcia projektowe (techniczno-informatyczne) -

- teoria projektów
 - o SMARTowy cel
 - o grupa docelowa
 - o budżet
 - o harmonogram
- psychologia grupy
 - o role w grupie
 - o umiejętności liderskie
 - o rozwiązywanie konfliktów
- w zależności od potrzeb grupy
 - o CAD
 - o druk 3D
 - o mechaniczne elementy handlowe
 - o obsługa podstawowych elektronarzędzi

ZS- Zajawka Specjalizacji -

Przedmiot ma na celu pokazać różne dziedziny poruszane na specjalizacji techniczno-informatycznej poprzez wykonanie projektu. Ważnym celem zajęć jest pokazanie różnorodności tematycznej, ale również przymusu łączenia hardware'u i software'u na zajęciach w klasach III i IV.

Klasa III

PPiW - Podstawy Projektowania i Wytwarzania -

- projektowanie 3D w Fusion 360
 - o podstawowe operacje - sketch, extrude, revolve, pattern, mirror, fillet, chamfer, operacje blachowe
 - o złożenia, wiązania
 - o zaawansowane operacje - sweep, loft, rib, web, emboss, shell
- podstawowe zagadnienia inżynierskie -
 - o rodzaje obróbki - druk 3D, skrawanie - praktyczne ćwiczenia
 - o omówienie innych metod (nieдоступnych w szkole) - cięcie laserem, gięcie blach, toczenie, spawanie
 - o rodzaje i zastosowanie łożysk, rodzaje napędów, elementy złączne, przekładnie
- realizacja wybranego projektu w grupach
 - o w tym roku - ploter 2D do rysowania markerem ścieżek PCB, mechaniczny pająk, samochód RC z elementami mechanicznymi drukowanymi w 3D

PZ - Programowanie zaawansowane 1 -

- Wymagania wstępne:
 - o Umiejętność programowania w jednym z języków: C++, Java, JavaScript, PHP

- Podstawowa wiedza o programowaniu obiektowym (znajomość pojęć: klasa, obiekt, hierarchia klas, dziedziczenie,)

Przedmiot „programowanie zaawansowane 1” ma zaznajomić uczniów z trzema zagadnieniami: programowaniem obiektowym, tworzeniem aplikacji okienkowych (desktopowych) oraz złożonymi algorytmami. Te trzy zagadnienia przekazywane będą równolegle poprzez projektowanie i implementowanie prostych aplikacji okienkowych (2-4 zajęcia) rozwiązujących wybrane praktyczne problemy np. matematyki, fizyki, chemii, biologii, itp. Jako środowisko uruchomieniowe proponowana jest platforma .NET i środowisko Visual Studio. Oprócz zaznajomienia się z poznawanymi zagadnieniami, zwracana będzie również uwaga na poprawność kodowania, dokumentowanie kodu, tworzenie podstawowej specyfikacji projektowanej aplikacji, używanie systemu kontroli wersji, itp.

Projektowanie i programowanie obiektowe

Celem jest wyrobienie umiejętności oraz zdobycie biegłości w programowaniu w jednym z popularnych języków obiektowych z wykorzystaniem wielu, często również zaawansowanych, konstrukcji tego języka (jako przykładowy język obiektowy proponowany jest język C#). Wybór języka programowania wynika z proponowanej platformy .NET, natomiast własności programowania obiektowego (paradygmaty) są wspólne dla wszystkich języków obiektowych C++, Java i częściowo PHP. Po ukończeniu przedmiotu uczniowie powinni: znać, rozumieć i umieć stosować w praktyce pojęcia i koncepcje paradygmatu programowania obiektowego takie jak klasa, klasa abstrakcyjna, interfejs, obiekt, dziedziczenie, polimorfizm, hermetyzacja danych

Programowanie w środowisku graficznym

Celem jest zaznajomienie uczniów z technologiami tworzenia aplikacji wykorzystujących graficzny interfejs użytkownika systemów MS Windows (w środowisku .NET). Po ukończeniu przedmiotu uczniowie powinni: posiadać wiedzę na temat ogólnego schematu działania aplikacji wykorzystujących okienkowy interfejs użytkownika, posiadać podstawową wiedzę o środowisku .NET, potrafić posłużyć się biblioteką Windows.Forms platformy .NET w celu utworzenia prostych aplikacji okienkowych dla systemu MS Windows.

Algorytmy

Celem jest zdobycie umiejętności konstruowania wydajnych algorytmów i dobierania właściwych struktur danych dla rozpatrywanych zagadnień, a także zapoznanie się z bardziej zaawansowanymi technikami konstruowania algorytmów jak programowanie dynamiczne, algorytmy z powrotami, algorytmy zachłanne, zasada "dziel i zwyciężaj". Celem jest również zapoznanie się z wydajnymi algorytmami dotyczącymi różnych dziedzin np. teorii grafów, grafiki komputerowej 2D, teorii baz danych, uczenia maszynowego, metod sztucznej inteligencji, matematyki, fizyki.

Program przedmiotu - semestr 1 (2h/tyg.)

Platforma .NET, C#, Aplikacje desktopowe

- Platforma .NET, historia, własności, wersje,
- Język C#, jako podstawowy obiektowy język programowania w .NET
- Podstawowe i zaawansowane elementy programowania obiektowego
 - Klasa, Obiekt, atrybuty, metody, konstruktory
 - Paradygmaty programowania obiektowego:
 - abstrakcja danych, enkapsulacja danych, dziedziczenie, polimorfizm
 - Klasy statyczne, abstrakcyjne, Interfejsy
 - Akcesory danych (getter i setter)
- Programowanie w środowisku graficznym (aplikacje desktopowe) - Windows Forms (.NET)
 - Pętla i obsługa zdarzeń
 - Typowe kontrolki UI
- Typowe aplikacje wspólnie realizowane na zajęciach
 - Prosta aplikacja konsolowa (Person/Student/Teacher) demonstrująca język C# i własności obiektowe
 - Aplikacja graficzna - rzut ukośny
 - Aplikacja graficzna – Gra w życie (tablice w C#)

- Aplikacja graficzna – całkowanie metodą Monte Carlo
- Aplikacja graficzna – wieże Hanoi (rekurencja)
- Aplikacja graficzna – Labirynt (edycja, szukanie wyjścia – algorytmy grafowe)
- Przykładowe własne projekty aplikacji graficznych
 - Gra „Reversi/Othello”
 - Gra „Saper”
 - Gra „MasterMind”
 - Gra „Kółko i Krzyżyk” na dużej planszy
 - Gra w "pary" (memory)
 - Gra typu „Wąż”
 - Gra „warcaby”
 - Symulacja ruchów i zderzeń kul na stole bilardowym
 - Statki – wstępne ułożenie statków na planszy
 - Labirynt – szukanie drogi
 - Symulacja podróży statku kosmicznego na księżyc/mars

Program przedmiotu - semestr 2 (2h/tyg.)

Platforma .NET c.d., algorytmy zaawansowane

- Komunikacja sieciowa
- Obsługa baz danych
- Programowanie synchroniczne i asynchroniczne (C#/Python/...)
- Wybrane zagadnienia zaawansowanych algorytmów z:
- zagadnień grafowych
 - uczenia maszynowego (ML)
 - metod sztucznej inteligencji (AI)
 - grafiki komputerowej 3D
- Typowe aplikacje wspólnie realizowane na zajęciach
 - Proste aplikacje konsolowe/desktopowe z obsługą baz danych, sieci, async/await
 - Aplikacja graficzna „rzut ukośny” z algorytmem genetycznym
 - Aplikacja ML – klasteryzacja oparta o metodę K-średnich
 - Aplikacja graficzna – Ray-casting dla kul
 - Aplikacja graficzna – renderer 3D
 - Aplikacja graficzna – VR
- Przykładowe własne projekty aplikacji ML, AI, 3D
 - Wyznaczanie optymalnej trasy przejazdu po torze - algorytm genetyczny
 - Renderowanie sceny algorytmem Ray-Tracing (rekurencja + model oświetlenia)
 - Znajdowanie najkrótszej drogi od/do algorytmem Dikstry
 - Implementacja prostej sieci neuronowej

Proj- Projekt -

Zajęcia mają na celu projektowe wykorzystanie zdobytej wiedzy poprzez budowę autonomicznej lub zdalnie sterowanej łodzi podwodnej. Projekty tworzone są w grupach 5-osobowych. Uczniowie i uczennice rozwijają zagadnienia związane z mechaniką, fizyką, robotyką, napędami, sterowaniem, komunikacją radiową i zarządzaniem projektem.

Rep - Repetytorium -

Zajęcia w zależności od potrzeb rocznika przyjmują różne formy. Może to być dodatkowy czas na rozwiązywanie zadań i problemów informatyczno-technicznych pod okiem doświadczonego dewelopera. Może to być również kontynuacja informatyki z naciskiem na system operacyjny Linux i język Python.

Klasa IV

HS - Hackerspace -

- podstawy technologii
 - podstawy rysunku technicznego
 - podstawy wymiarowania
- obsługa maszyn przemysłowych i elektronarzędzi
 - bezpieczeństwo i higiena pracy
 - ubiór
 - środki ochrony osobistej
 - ploter laserowy (drewno i plexi)
 - tokarka z DRO
 - frezarka CNC
 - 6-osiowy robot przemysłowy
 - wiertarka kolumnowa
 - spawarki MIG i TIG
 - przecinarka taśmowa
 - szlifierki, wiertarki, piły, dremel
- podstawowe operacje warsztatowe
 - podstawy metrologii
 - suwmiarka
 - mikrometr
 - gwintowanie
 - szlifowanie
 - fazowanie
- mechaniczne elementy handlowe
 - śruby, wałki, prowadnice, łożyska
- zagadnienia społeczne
 - planowane postarzanie przedmiotów
 - naprawa prostych usterek
- projekty własne z wykorzystaniem zdobytej wiedzy i nabywanej nowej na bieżąco

PZ2 - Programowanie zaawansowane 2 -

Program nauczania

Przedmiot „programowanie zaawansowane 2” ma zaznajomić uczniów z zaawansowanymi metodami i technologiami projektowania i projektowania nowoczesnych aplikacji webowych i mobilnych

Program przedmiotu - semestr 1 (2h/tyg.)

- architektura wielowarstwowa aplikacji webowych
- aplikacje „tradycyjne” (MPA) a aplikacje SPA (single page application)
- protokół REST API, Testowanie - Postaman
- podstawy autentykacji - JWT
- zabezpieczanie i typowe ataki na aplikacje webowe (sql injection, xss, csrf,...)
- programowanie aplikacji klienckiej (frontend), wybrane frameworki:
 - react, vue.js, flutter
- programowanie aplikacji serwera (backend), wybrane frameworki:
 - Python (flask, FastAPI, django), Php (Laravel), javascript (Node.js), .NET Core
- programowanie aplikacji mobilnych:
 - flutter, react native
- Przykładowe własne/wspólne projekty aplikacji webowej/mobilnej:
 - Aplikacja webowa/mobilna z komunikacją z IDU (poprzez API)
 - Projekt „OZE – Razem” (mapa farm wiatrowych + siła wiatru z API)
 - Wyświetlanie lokalizacji pojazdów na podstawie API ZTM

Rep - Repetytorium -

Zajęcia w zależności od potrzeb rocznika przyjmują różne formy. Może to być dodatkowy czas na rozwiązywanie zadań i problemów informatyczno-technicznych pod okiem doświadczonego dewelopera. Może to być również kontynuacja informatyki z naciskiem na system operacyjny Linux i język Python.