

### **3.7 Asztali és mobil alkalmazások fejlesztése, szoftvertesztelés és adatbázis-kezelés megnevezésű tanulási terület**

A tanulási terület tantárgyainak összóraszám:

603/603 óra

A tanulási terület tartalmi összefoglalója

A kódolás és a programozás képessége ma már nemcsak a szoftverfejlesztők számára nélkülözhetetlen, hanem az összes egyéb területen dolgozó informatikus számára is elengedhetetlen. A fentiek miatt fontos, hogy a közös alapozás jegyében minden tanuló megszeresse a kódolást, fejlett problémamegoldó és algoritmizáló képességre tegyen szert, és képessé váljon egyszerűbb problémák kezelését végző alkalmazások és szkriptek készítésére. Elengedhetetlen, hogy minden tanuló tudjon webes környezetben dolgozni, valamint magabiztosan kezelje az adatokat és az adatbázisokat. A tanulási terület oktatásának fontos feladata az is, hogy a tanulókat megtanítsa csapatban dolgozni, valamint a közös munkát segítő forráskódkezelők és a csoportmunkát segítő online eszközök kezelésére.

A tanulási terület az alábbi főbb témákat érinti:

- Haladó és objektumorientált programozás – továbblépés a programozás területén egy második nyelv (C# vagy Java) segítségével
- Adatbázis-kezelés – a relációs adatbázisokkal és SQL-lekérdezésekkel kapcsolatos legfőbb készségek elsajátítása
- Szoftvertesztelés

#### **3.7.1 Asztali alkalmazások fejlesztése tantárgy**

**180/180 óra**

##### **3.7.1.1 A tantárgy tanításának fő célja**

A tantárgy oktatásának alapvető célja azoknak az elméleti ismereteknek az átadása, valamint az ezekhez tartozó készségeknek a fejlesztése, amelyek képessé teszik a tanulót egy egyszerűbb alkalmazás programozására, a megvalósításhoz szükséges algoritmus elkészítésére, a szükséges adattípusok és adatszerkezetek kiválasztására. A tantárgy oktatásának fontos feladata az is, hogy a tanuló problémamegoldó készségét fejlessze. A tantárgy további célja, hogy a kapcsolódó ipari minősítések megszerzésére is felkészítse a tanulókat.

Az Asztali alkalmazások fejlesztése tantárgy segítségével a tanulók egy konkrét programozási nyelven keresztül magasabb szintű, mélyebb tudást sajátítanak el, mely jó alapot biztosít a későbbi, egyéb programozási nyelvek elsajátításához is. Oktatása C# vagy Java nyelv segítségével történhet, ennek a változatnak a célja a programozás magasabb szintű elsajátítása általános programozási és Java vagy C# nyelvspecifikus aspektusokra fókuszálva. A tantárgy, bár a Java vagy C# nyelv elemeire és annak sajátosságaira épít, mégis általános, széles körben elterjedt programozási logikát és problémamegoldást oktat, mely általános a modern programozási területeken.

Természetesen nagyon sok hasonló, a későbbi szakmai fejlődést is megalapozó, kiváló programozási környezet létezik, amelyek hasonlóan jó alternatívát jelentenének. A fenti programozási nyelvek a legszélesebb körben használtak közé tartoznak, megismerésük után a tanulók olyan általános készségekre tesznek szert, amelyekkel képesek lesznek a későbbi munkakörnyezetükben más programozási környezetek gyors elsajátítására és hatékony használatára. Természetesen a kerettanterv nem zárja ki, hogy a szaktanár, az előírt ismeretek átadásán túl, további szakmai ismereteket is átadjon, így például saját döntése alapján betekintést nyújthat más korszerű programozási környezetekbe is.

A tantárgy oktatása során az elméleti anyag tanítása közvetlenül a gyakorlati példákon keresztül valósulhat meg, az elméleti anyag kisebb részekben történő ismertetése és annak azonnali, gyakorlati példákon, feladatokon történő gyakorlása során.

Kétéves képzés esetén, 1/13. évfolyamon a tantárgy oktatása csak a Programozási alapok tantárgy oktatása után történik.

3.7.1.2 A tantárgyat oktató végzettségére, szakképesítésére, munkatapasztalatára vonatkozó speciális elvárások

—

3.7.1.3 Kapcsolódó közismereti, szakmai tartalmak

—

3.7.1.4 A képzés órakeretének legalább 80%-át gyakorlati helyszínen (tanműhely, üzem stb.) kell lebonyolítani.

### 3.7.1.5 A tantárgy oktatása során fejlesztendő kompetenciák

Készségek, képességek	Ismeretek	Önállóság és felelősség mértéke	Elvárt viselkedésmódok, attitűdök	Általános és szakmához kötődő digitális kompetenciák
Konzolos vagy egyszerű asztali ablakos programot készít (kódol, fordít és futtat).	Ismeri a fejlesztéshez szükséges eszközöket (editor és futtatókörnyezet).	Teljesen önállóan	Legyen fogékony az információk befogadására és alkalmazására. Érdeklődjön az adott téma iránt.	JAVA / C# programozási nyelv és legalább egy fejlesztői környezet használata
Egyszerű és közepesen összetett programozási feladatokat algoritmizál és kódol.	Alapvető algoritmuselméleti ismerettel rendelkezik, ismeri a nyelvspecifikus szintaxist.	Teljesen önállóan		Hatékony internetes keresés és részvétel a legfontosabb szakmai közösségi platformokon (pl. GitHub, Stack Overflow) tudás-szerzés és -megosztás céljából
Az objektumorientált alapelveket betartva beépített osztályokat használ problémamegoldásra.	Ismeri az objektumorientált szintaxist és a beépített függvénykönyvtárakat.	Teljesen önállóan		

Olyan asztali grafikus alkalmazásokat készít, amelyek egyszerű interakciókat tesznek lehetővé a felhasználó számára, eseményvezérelt kódrészletekkel.	A JavaFX vagy a WPF fejlesztői környezet ismerete	Teljesen önállóan		
---	---	-------------------	--	--

### 3.7.1.6 A tantárgy témakörei

#### 3.7.1.6.1 Bevezetés a szoftverfejlesztésbe

A témakör célja, hogy átfogó ismeretet adjon a diákoknak a modern szoftverfejlesztés általános lépéseiről, és a Java/C# programozási nyelvnek az iparban világviszonylatban betöltött jelenkori helyzetéről. Feladata továbbá a programozási környezet megismertetése, a projektlétrehozás és egyéb előkészületi feladatok megismertetése, ami szükséges a későbbi, önálló programozási feladatokhoz.

A témakörben az alábbi ismeretek és gyakorlati készségek elsajátítására kerül sor:

- A Java/C# nyelv története, kialakulása és fejlődése
- A Java/C# nyelvcsalád elemei
- Forrás- és konfigurációs fájlok, futtatókörnyezetek, frameworkök
- A fejlesztőkörnyezet megismerése
- Kiinduló szintaxis konzolos és grafikus alkalmazás esetén
- Szoftverfejlesztési paradigmák, spirális fejlesztés, folyamatos fejlesztés
- Kódsorok olvasása, utasítások értelmezése, debugolási alapok (breakpoints, lépétesítés)
- Utasítások lezárása, a whitespace karakterek hatása
- Kódolási stílus bevezető

#### 3.7.1.6.2 Procedurális és objektumorientált szoftverfejlesztés

A témakör célja, hogy megadja azt a minimálisan szükséges alapot, amellyel egy Java/C# program fejlesztéséhez neki lehet kezdeni. A témakörök a későbbiekben részletesen tárgyalják még az objektum-orientált szemléletet, itt csak azt az alapot sajátítják el a diákok, amely segítségével megérthetik a programszerkezet és a segédkönyvtárak alapvető működését, és fel tudják azt használni. Saját osztály megírása a későbbi anyag rész feladata lesz.

A témakörben az alábbi ismeretek és gyakorlati készségek elsajátítására kerül sor:

- A procedurális és az objektumorientált szemlélet összevetése
- Az osztály és az objektum fogalma, a köztük lévő különbségek
- Példányosítás
- Property-k
- Java/C# szintaxisban kifejezve az objektumorientáltság

#### 3.7.1.6.3 Változók

A témakör célja, hogy megismertesse a tanulókkal a különböző változók típusait, felhasználásuk módját, és a hozzájuk kapcsolódó szintaxist és műveleteket. A témakör a változokon értelmezett műveleteken felül kitér a változók tartalmának különféle inicializálására és változtatására, a fájlbeolvasásra, a gyakran használt művelettípusokra (pl. Stringkezelés, fájlbeolvasás), de a segédkönyvtárak teljes körű használatára, valamint a bonyolultabb műveletekre még nem tér ki. A boolean típus bevezetése is későbbre, a vezérlési szerkezetek

témakörhöz kerül, hogy a diákok közvetlenül azok gyakorlati fontosságának kontextusában tanulhassák meg használatukat.

A témakörben az alábbi ismeretek és gyakorlati készségek elsajátítására kerül sor:

- Bevezetés
- Elméleti bevezetés
- Primitívek: boolean, int, double és String
- Elnevezési konvenciók
- Numerikus típusok
- Integer típusú primitívek és köztük lévő különbségek: byte, short, int, long
- Lebegőpontos típusok: float, double
- Aritmetikai, és matematikai műveletek
- Zárójelezés
- Szöveges típusok
- Char típus
- Stringkezelési alapok
- String konkatenáció
- Whitespace és escape karakterek
- Konzolra írás
- Típuskonverzió
- Automatic promotion
- Type casting
- Számérték-konverzió
- Stringek parse-olása
- Inputról olvasás
- User input beolvasása konzolról
- User input beolvasása grafikus felületről
- Fájlbolvasási alapok

#### **3.7.1.6.4 Metódusok**

A témakör célja, hogy megismertesse a diákokkal a metódusok/függvények szintaxisát és gyakorlati alkalmazásukat.

A témakörben az alábbi ismeretek és gyakorlati készségek elsajátítására kerül sor:

- Függvénydefiníciók
- Kód strukturálása osztályszinten
- Függvények haszna és célja
- Osztályváltozók elérése
- Scope
- Argumentumok, visszatérési érték

#### **3.7.1.6.5 Beépített segédosztályok**

A témakör célja, hogy megismertesse a nyelv részét képző és gyakori programozási problémák megoldásához eszközt biztosító segédkönyvtárak használatát. A segédkönyvtárak használatával, mely minden programozási nyelvben általános gyakorlat, a kód-újrafelhasználás és a külső eszközök használatának hasznosságát hivatott példázni.

A témakörben az alábbi ismeretek és gyakorlati készségek elsajátítására kerül sor:

- String osztály
- Nyelvi beépített osztályok használatának alapja, dokumentációértelmezés
- String osztály legfontosabb függvényei és használatuk: indexOf, charAt, substring()
- A Stringkonkatenálás és az immutable fogalma
- Stringek összehasonlítása

- A Random osztály
- Véletlenszám-generálás felhasználása a programozásban, pszeudo-véletlen elméleti kitekintő
- Véletlenszám generálása a Random osztály segítségével, next...() függvények
- A Math osztály
- Math osztály felhasználásának lehetőségei
- Statikus metódusok szerepe
- Legfontosabb függvények: max(), min(), sqrt()

#### 3.7.1.6.6 Vezérlési szerkezetek, ciklusok

A témakör célja, hogy megismertesse a Bool típusú logikai változókat és az alapvető programozási vezérlési szerkezeteket, a használt szintaxisaikkal együtt. Kitér a különböző elágazásokra, felhasználási módjaikra, vezérlési szerkezetekre és gyakorlati hasznosságukra. Ezenkívül a tanulók elsajátítják a programozásban használt általános ciklusok típusait és azok felhasználási módjait.

A témakörben az alábbi ismeretek és gyakorlati készségek elsajátítására kerül sor:

Logikai (boolean) kifejezések

- Boolean változók értelmezése, inicializálása, deklarálása
- Aritmetikai alapfogalmak, boolean változók összehasonlítása, operátorok
- Két- és többirányú (if-then-else) elágazás
- Döntési változók, vezérlési struktúrák elméleti bevezető
- Feltételeken alapuló futtatás
- If-then-else elágazások

Összetett kifejezések, magas szintű operátorok:

- Hármass operátor (? : )
- Érték szerinti (Switch) elágazás
- Switchelmélet
- Szintaxis
- Összehasonlítás If-fel

Ciklusok:

- “for” ciklus
- Ciklusok jelentősége elméleti bevezető
- A for ciklust ismertető elmélet
- For ciklus szintaxis
- Ciklus scope
- Ciklusok debugolása
- Végtelen ciklus
- “do-while” ciklusok
- Elöltesztelő ciklus
- Hátultesztelő ciklus
- Ciklustípusok összehasonlítása, mikor melyiket érdemes használni

Ciklusvezérlés:

- Ciklus futtatásának leállítása, Break
- Ciklusátugrás, Continue

#### 3.7.1.6.7 Tömbök és listák

A témakör célja, hogy megismertesse a tömbök és listák adatszerkezet-definícióját, felhasználását. A témakör az egyszerű tömbökön túl a lista segédosztályok segítségével rámutat a listák használatának előnyeire, hátrányaira.

A témakörben az alábbi ismeretek és gyakorlati készségek elsajátítására kerül sor:

- Egydimenziós tömb
- Tömbök bevezetése
- Tömb definiálása, inicializálása
- Tömb elemének módosítása
- Tömb bejárása ciklusok segítségével
- Hibakezelés, túlcímzés
- Többdimenziós tömb
- Mátrixok kezelése
- Listák bevezetése
- ArrayList / List definiálása, inicializálása
- ArrayList / List elemének módosítása
- ArrayList / List bejárása iterátor- és ciklusok segítségével
- Autoboxing

#### **3.7.1.6.8 Kivételkezelés, hibakeresés**

A témakör célja, hogy megismertesse a tanulókat a kivétel (exception) fogalmával, annak működési elvével és speciális vezérlési struktúrájának sajátosságaival, valamint bevezesse a diákokat a programozási nyelvben használatos hibakeresési módszerek világába, és felhasználható eszköztárat nyújtson a különféle hibatípusok azonosítására és javítására.

Az „opcionális” jelzéssel szereplő ismeretek tárgyalása nem kötelező, a kapcsolódó szakmai vizsga ezeket nem kéri számon. A témakörben az alábbi ismeretek és gyakorlati készségek elsajátítására kerül sor:

Kivételkezelés:

- Kivételkezelés elméleti bevezető
- Try-catch szintaxis
- Beépített kivételtípusok

Hibakeresés (opcionális):

- Programkód tesztelése és hibakeresés
- A hibák 3 osztályának azonosítása (fordítási idejű, logikai, futási idejű)
- Hibakezelési technikák: print, IDE, kifejezésszerkesztő, előnyök-hátrányok
- Leggyakoribb hibák

#### **3.7.1.6.9 Objektumorientált fejlesztés**

A témakör célja, hogy mélyebb, átfogó tudást adjon át OOP témakörben, mely segítségével a tanulók saját osztályokat képesek létrehozni, és objektumorientált szemlélet segítségével feladatokat megoldani.

A témakörben az alábbi ismeretek és gyakorlati készségek elsajátítására kerül sor:

Osztályok

- OOP-projekt létrehozása, osztályok létrehozása IDE segítségével
- Osztályszintű scope
- Osztályváltozók és -mezők publikussága
- Érték szerinti átadás függvényhívások esetén
- Visszatérési érték
- Main függvény

Példányosítás

- Esettanulmány bevezetése
- Esettanulmány osztály létrehozása, osztályok példányosításának szintaxisa
- New kulcsszó
- Objektumok, objektumok összehasonlítása

- Objektumreferenciák
- Memóriakezelési alapok

#### Konstruktorok

- Konstruktorok működése, szintaxis
- Default érték, nullérték
- Osztályváltozók inicializálásának fontossága
- Osztályváltozók közvetlen módosítása
- Osztályváltozók módosítása konstruktor segítségével, konstruktorszabályok
- Default konstruktor
- This kulcsszó

#### **3.7.1.6.10** Grafikus programozás

A témakör célja, hogy bevezesse a tanulókat az egyszerű (asztali) grafikus alkalmazás alapjaiba. A témakör célja, hogy a tanulók létre tudjanak hozni egyszerű, ablakos alkalmazásokat, melyekre a leggyakoribb eszközöket (gomb, beviteli mező, label stb.) képesek felhelyezni, és ezekre egyszerű, eseménykezelő programrészleteket, metódusokat megfogalmazni.

A témakörben az alábbi ismeretek és gyakorlati készségek elsajátítására kerül sor:

##### Bevezetés:

- GUI programozási alapok
- JavaFX / WPF program szerkezete, létrehozása
- JavaFX / WPF komponensek

##### Színek és formák:

- JavaFX / WPF színkezelés
- JavaFX / WPF shape-ek, property-k, és működési minták
- JavaFX / WPF egyszerűbb sokszögek kirajzolása
- JavaFX / WPF Image és ImageView

##### Eseményvezérelt szoftverfejlesztés:

- Eseményvezérelt programozás, elmélet
- JavaFX / WPF gomb, beviteli mező
- JavaFX / WPF eseményfigyelés, aszinkron kód, szintaxis