

SZAKÉRTŐI VÉLEMÉNY

Képzés (képzési program) megnevezése	Junior vállalati Java backend fejlesztő
Felnőttképző megnevezése és engedélyszáma:	Training360 Kft. E/2020/000016
Szakértői megállapítások	
<ol style="list-style-type: none">1. A képzési program tartalma megfelel a felnőttképzésről szóló 2013. évi LXXVII. törvénynek és szakmai oktatás vagy szakmai képzés esetén a szakképzésről szóló törvénynek és a szakképzésről szóló törvény végrehajtásáról szóló kormányrendeletnek.2. A képzési programban meghatározott tartalommal, feltételekkel és módon, valamint a képzéssel érintett célcsoport számára megszerezhetők a képzési programban megjelölt kompetenciák.3. A képzési program minden oldala folyamatos oldalszámozással van ellátva, és az összefűzésre úgy került sor, hogy annak szétválasztására sérülésmentesen nincs lehetőség.	
Szakértői vélemény kelte	Budapest, 2021.06.04.
Felnőttképzési szakértő neve, nyilvántartási száma	Pádár Tivadar FSZ/2020/000057
Felnőttképzési szakértő aláírása	

KÉPZÉSI PROGRAM

SZAKMAI KÉPZÉS

JUNIOR VÁLLALATI JAVA BACKEND FEJLESZTŐ

(PROGRAMKÖVETELMÉNY AZONOSÍTÓ: 06135011)

Vonatkozó jogszabályok:

2019. évi LXXX. törvény a szakképzésről
12/2020. (II. 7.) Korm. rendelet
2013. évi LXXVII. törvény
11/2020. (II. 7.) Korm. rendelet



1. Alapadatok

A programkövetelmény alapján szervezhető szakmai képzés:	
1.1.	Megnevezése: Junior vállalati Java backend fejlesztő
1.2.	Ágazat megnevezése: Informatika és távközlés ágazat
1.3.	Besorolása a képzési területek egységes osztályozási rendszere (KEOR) szerinti kód alapján: 0613 Szoftverek és alkalmazások fejlesztése és elemzése
A programkövetelmény alapján szervezhető szakmai képzéssel megszerezhető szakképesítés:	
1.4.	Megnevezése: Junior vállalati Java backend fejlesztő
1.5.	Az Európai Képesítési Keretrendszer (EKKR) szerinti szint: 5
1.6.	A Magyar Képesítési Keretrendszer (MKKR) szerint szint: 5
1.7.	A Digitális Kompetencia Keretrendszer szerinti szint: 6
1.8.	<p>A programkövetelmény alapján szervezhető szakmai képzéssel megszerezhető szakképesítés és az azzal betölthető munkakör vagy végezhető tevékenység kapcsolata, összefüggése:</p> <p>A szakmai képzéshez kapcsolódóan megszerezhető szakképesítéshez szükséges kompetenciákkal szakmajegyzékben szereplő szakma körébe vonható munkaterület, tevékenység vagy munkakör magasabb szinten gyakorolható, vagy a szakmai képzés szakmajegyzékben szereplő szakma képzési és kimeneti követelményeiben meg nem határozott speciális szakmai ismeretek és szakmai készségek megszerzésére irányul.</p> <p>Junior vállalati Java backend fejlesztő Java programozási nyelven önálló háromrétegű backend alkalmazásokat fejleszt perzisztens réteggel, melyhez valamilyen ORM technológiát használ, és REST API-val. Az alkalmazáshoz unit és integrációs teszteseteket ír. Ennek megvalósításához valamilyen nagyvállalati keretrendszert használ, mint pl. Spring Framework/Spring Boot vagy Java EE implementáció. Munkája elvégzéséhez használ valamilyen konténerizációs technológiát, pl. Docker.</p>
1.9.	<p>A képzés célja:</p> <p>A képzés célja, hogy a képzésben résztvevő sajátítsa el a Junior vállalati Java backend fejlesztő szakképesítés megszerzéséhez szükséges elméleti és gyakorlati ismereteket, készségeket és kompetenciákat.</p> <p>A képzés célja, hogy a résztvevők önállóan képesek legyenek felépíteni egy többretegűbackend alkalmazást valamilyen nagyvállalati backend keretrendszerrel (String/Java EE). Megismerjék a különböző perzisztens technológiákat, mint pl. a JPA. Valamint megismerjék és elsajátítsák, hogyan kell REST API-t fejleszteni. Az alkalmazás teszteléséhez megismerjék és elsajátítsák a unit és integrációs tesztelés fogalmait és módszereit.</p>
1.10.	<p>A képzés célcsoportja:</p> <p>A képzés célcsoportját jelenti minden olyan személy, aki a belépési feltételeknek megfelel és a képzési programmal elérhető ismeretek, készségek és kompetenciák megszerzését tűzte ki célként maga elé. A képzés célcsoportja olyan junior Java fejlesztők, akik tovább szeretnének fejlődni, és bonyolultabb backend alkalmazások fejlesztését szeretnék elsajátítani.</p>
1.11.	<p>Megszerezhető kompetenciák:</p> <ul style="list-style-type: none"> • Verziókezelés Git verziókövető rendszerrel • Unit és integrációs tesztelés JUnit, Mockito és AssertJ használatával • Clean Code alapelvek megismerése • Backend alkalmazások fejlesztése Spring Boot keretrendszerrel • Adatbázis és Spring Boot alkalmazás futtatása Docker konténerben • REST API Spring MVC-vel • REST API Swaggerrel • Tesztelés RestTemplate használatával • Adatbáziskezelés JPA használatával (ORM) • Adatbáziskezelés Spring Data JPA használatával • Séma inicializálás Flyway vagy Liquibase használatával • Konténerizáció Dockerrel

2. A képzésbe való bekapcsolódás és részvétel feltételei

2.1.	Iskolai végzettség	Középfokú végzettség
2.2.	Szakmai végzettség	-
2.3.	Szakmai gyakorlat	-
2.4.	Egészségügyi alkalmasság	-
2.5.	Előzetesen elvárt ismeretek	A 06134005 Junior Java backend fejlesztő szakképesítésben meghatározott szakmai kompetenciák megléte a végzettségről szóló tanúsítvánnyal, vagy ennek hiányában, a képző által szervezett előzetes szintfelmérő vizsga sikeres teljesítésével bizonyítottan.
2.6.	Egyéb feltételek	Minden résztvevővel az Fktv. (a felnőttképzésről szóló 2013. évi LXXVII. törvény és végrehajtási rendelete (a 11/2020. (II. 7.) Korm. rendelet) által meghatározott tartalmú felnőttképzési szerződést kell kötni. A felnőttképzési szerződésben feltüntetésre kerül az esetleges előzetes tudásmérés eredménye, továbbá az esetleges előzetesen megszerzett tudás beszámítása.

3. Tervezett képzési idő

3.1.	A képzés óraszám:	320 óra
3.2.	Megengedett hiányzás mértéke:	A képzés teljes idejének (320 óra) megengedett hiányzás mértéke 20% (64 óra kontaktóra esetén). <i>(A felnőttképzési szerződést azonnali hatállyal felmondható, ha a képzésben részt vevő személy a kontaktórákról a képzési programban meghatározott időnél többet mulasztott.)</i>

4. Tananyagegységek

A képzés tananyagegységeinek megnevezése:	Óraszám:
Unit tesztelés JUnit eszközzel	40 óra
Backend alkalmazás fejlesztése, integrációs tesztelése Spring Boot keretrendszerrel, futtatás Dockerrel	120 óra
Adatbáziselérés JPA és Spring Data JPA használatával	80 óra
Projektmunka	80 óra

4.1. Tananyagegység

4.1.1.	Megnevezése:	Unit tesztelés JUnit eszközzel
4.1.2.	Célja:	A tananyagegység célja, hogy a résztvevő megismerje a JUnit, Mockito és AssertJ eszközöket. Ezek használatával megfelelő unit tesztek tudjon megvalósítani.
4.1.3.	Megvalósítása során alkalmazott munkaformák:	Frontális oktatás, mentor által felügyelt egyéni- és csoportmunka, távoktatás
4.1.4.	Megvalósítása során alkalmazott képzési módszerek:	Előadás, magyarázat, szemléltetés, gyakorlat, megbeszélés, együttes és önálló tananyag feldolgozás, önálló tanulás, csoportos feladatmegoldás
4.1.5.	Óraszám:	40 óra

4.1.6.	Beszámítható óraszám ¹ :	40 óra		
4.1.7.	A programkövetelményben meghatározott, a tananyagegység keretében elsajátítandó szakmai követelmények:			
S.sz.:	Készségek, képességek:	Ismeretek:	Elvárt viselkedésmódok, attitűdök:	Önállóság és felelősség mértéke:
1.	Az alkalmazáshoz, melyet fejleszt, unit és integrációs tesztek írását és futtatását valamely keretrendszer használatával (JUnit, TestNG, stb.).	Ismer legalább egy unit és integrációs teszt eszközt (JUnit, TestNG, stb.), ismeri azok képességeit, funkcióit.	Elkötelezett abban, hogy tesztekkel lefedje az általa készített kódot a használt teszteszköz funkcióinak kihasználásával.	Önállóan teszt eszközt választ és teszteseteket ír, melyekkel a funkcionális helyesség biztosítható és automatizáltan újra tesztelhető.
	Megtanítandó és elsajátítandó témakör(ök); kapcsolódó foglalkozás(ok) megnevezése, (óra)száma és tartalmi elemei:			
	Témaköre: Unit tesztelés (10 óra) Foglalkozás neve: Unit tesztelés (10 óra) Tartalmi elemei: <ul style="list-style-type: none"> • Unit tesztelés • Tesztesetek életciklusa • Tesztesetek ismétlése 			
S.sz.:	Készségek, képességek:	Ismeretek:	Elvárt viselkedésmódok, attitűdök:	Önállóság és felelősség mértéke:
2.	Megfelelően képes használni egy verziókezelő rendszert a hatékony csoportmunkához (Git, Subversion, TFS, stb.)	Ismeri a verziókövető rendszer képességeit, parancsait.	A lehető leghatékonyabban vesz részt a csoportmunkában a saját munkájának megosztásával.	Önállóan használja a verziókövető rendszert saját munkájának publikálására.
	Megtanítandó és elsajátítandó témakör(ök); kapcsolódó foglalkozás(ok) megnevezése, (óra)száma és tartalmi elemei:			
	Témaköre: Verziókezelés Gittel (10 óra) Foglalkozás neve: Verziókezelés Gittel (10 óra) Tartalmi elemei: <ul style="list-style-type: none"> • Git felépítése • Git parancsok • Csoportmunka Git használatával • Publikálás GitHubra 			
S.sz.:	Készségek, képességek:	Ismeretek:	Elvárt viselkedésmódok, attitűdök:	Önállóság és felelősség mértéke:
3.	A teszteseteket képes különböző környezetekben futtatni (akár fejlesztőeszközben, mint IntelliJ IDEA, Eclipse, NetBeans, stb., akár build eszköz segítségével,	Ismeri azokat a lehetőségeket, hogy hogyan lehet futtatni a unit és integrációs tesztek.	Törekszik arra, hogy a teszteseteket a fejlesztés közben a fejlesztőeszközből is futtassa, valamint build eszközből, akár parancssor használatával.	Önállóan futtat teszt eseteket a fejlesztőeszközből és a parancssorból.

¹ Kontaktórától eltérő munkaforma alkalmazása esetén, ha az a képzés óraszámába beszámítható

	mint Maven, Gradle, stb.)			
	Megtanítandó és elsajátítandó témakör(ök); kapcsolódó foglalkozás(ok) megnevezése, (óra)száma és tartalmi elemei:			
	Témaköre: Futtatás IDE-vel és Mavennel (5 óra) Foglalkozás neve: Futtatás IDE-vel és Mavennel (5 óra) Tartalmi elemei: <ul style="list-style-type: none"> • Futtatás fejlesztőeszközből (IntelliJ IDEA) • Futtatás Mavennel • Tesztlefedettség 			
S.sz.:	Készségek, képességek:	Ismeretek:	Elvart viselkedésmódok, attitűdök:	Önállóság és felelősség mértéke:
4.	A tesztesetekben képes olvasható és hatékony ellenőrzéseket (assert) implementálni (pl. JUnit, TestNG, Hamcrest, AssertJ, stb.).	Ismert egy eszközt, mellyel könnyen karbantartható assert kifejezéseket tud írni.	Törekszik, hogy röviden és tömören fogalmazza meg az ellenőrzési kifejezéseket (assert) valamilyen erre specializált keretrendszer használatával.	Önállóan ír assert kifejezéseket, melyek olvashatóbbá teszik az ellenőrzéseket a tesztesetekben.
	Megtanítandó és elsajátítandó témakör(ök); kapcsolódó foglalkozás(ok) megnevezése, (óra)száma és tartalmi elemei:			
	Témaköre: Assertek (10 óra) Foglalkozás neve: Assertek (10 óra) Tartalmi elemei: <ul style="list-style-type: none"> • JUnit assertek • AssertJ használata • AssertJ kiterjeszthetősége 			
S.sz.:	Készségek, képességek:	Ismeretek:	Elvart viselkedésmódok, attitűdök:	Önállóság és felelősség mértéke:
4.	Unit tesztelésnél a kapcsolódó komponenseket mockolja erre használatos keretrendszerrel (pl. Mockito, PowerMock, stb.)	Ismert egy mock keretrendszert, mellyel a kapcsolódó komponenseket tudja kicserélni tesztelésre előkészített párjokkal (test double).	Törekszik arra, hogy a unit tesztek esetén semmilyen valós másik komponenst ne használjon, hiszen az már integrációs teszt. Erre valamilyen keretrendszert használ.	Önállóan helyettesíti a kapcsolódó komponenseket azok tesztelésre előkészített párjokkal.
	Megtanítandó és elsajátítandó témakör(ök); kapcsolódó foglalkozás(ok) megnevezése, (óra)száma és tartalmi elemei:			
	Témaköre: Mockolás (5 óra) Foglalkozás neve: Mockolás (5 óra) Tartalmi elemei: <ul style="list-style-type: none"> • Mockolás elmélete • Mockito 			

4.2. Tananyagegység

4.2.1.	Megnevezése:	Backend alkalmazás fejlesztése, integrációs tesztelése Spring Boot keretrendszerrel, futtatás Dockerrel		
4.2.2.	Célja:	A tananyagegység célja egy Spring Boot alkalmazás felépítése, REST API-val, annak unit és integrációs tesztelése, valamint futtatása Docker konténerben.		
4.2.3.	Megvalósítása során alkalmazott munkaformák:	Frontális oktatás, mentor által felügyelt egyéni- és csoportmunka, távoktatás		
4.2.4.	Megvalósítása során alkalmazott képzési módszerek:	Előadás, magyarázat, szemléltetés, gyakorlat, megbeszélés, együttes és önálló tananyag feldolgozás, önálló tanulás, csoportos feladatmegoldás		
4.2.5.	Óraszám:	120 óra		
4.2.6.	Beszámítható óraszám ² :	120 óra		
4.2.7.	A programkövetelményben meghatározott, a tananyagegység keretében elsajátítandó szakmai követelmények			
S.sz.:	Készségek, képességek:	Ismeretek:	Elvárt viselkedésmódok, attitűdök:	Önállóság és felelősség mértéke:
1.	Klasszikus CRUD műveleteket megvalósító API végpontokat fejleszt (pl. REST webszolgáltatások Spring Framework Controllerek, vagy JAX-RS implementáció használatával, stb.). Ezzel felépíti a prezentációs réteget.	Ismer legalább egy vállalati backend technológiát (Spring Controller, JAX-RS, stb.), és hogy hogyan kell vele API-t fejleszteni. Tisztában van az API tervezéssel, van megoldása CRUD műveletek elkészítésére, hibakezelésre, validációra.	Törekszik a szabványoknak megfelelő API kialakítására (pl. OpenAPI szabvány, stb). Olyan API-t alakít ki, melyet felületi technológia (pl. JavaScript keretrendszer, mobil vagy vastag kliens) tud használni.	Önállóan alakítja ki az alkalmazás API felületét. Dönt a hibakezelés és validáció megvalósításáról.
Megtanítandó és elsajátítandó témakör(ök); kapcsolódó foglalkozás(ok) megnevezése, (óra)száma és tartalmi elemei:				
Témaköre: REST API fejlesztése (30 óra) Foglalkozás neve: REST API fejlesztése (30 óra) Tartalmi elemei: <ul style="list-style-type: none"> • Spring Beanek • REST: GET, POST, PUT, DELETE • Státusz kódok és hibakezelés • Swagger • RestTemplate • Validáció 				
S.sz.:	Készségek, képességek:	Ismeretek:	Elvárt viselkedésmódok, attitűdök:	Önállóság és felelősség mértéke:
2.	Alkalmazásokat és az adatbázist konténerizált környezetben futtat (pl. Docker).	Ismer egy konténerizációs technológiát és hogy hogyan kell konténereket	Törekszik arra, hogy az alkalmazásait úgy írja meg, hogy azok konténerizált környezetben is futtathatóak legyenek.	Önállóan olyan tervezési döntéseket hoz, hogy az alkalmazás konténerizált környezetben is futtatható legyen.

² Kontaktórától eltérő munkaforma alkalmazása esetén, ha az a képzés óraszámába beszámítható

		létrehozni és kezelni. Ismeri annak módját, hogy hogyan csomagolja az alkalmazást valamilyen konténerbe.		
Megtanítandó és elsajátítandó témakör(ök); kapcsolódó foglalkozás(ok) megnevezése, (óra)száma és tartalmi elemei:				
Témaköre: Docker (20 óra) Foglalkozás neve: Docker (20 óra) Tartalmi elemei: <ul style="list-style-type: none"> • Docker alapfogalmak • Docker image készítés • Docker futtatás 				
S.sz.:	Készségek, képességek:	Ismeretek:	Elvárt viselkedésmódok, attitűdök:	Önállóság és felelősség mértéke:
3.	Az alkalmazáshoz, melyet fejleszt, unit és integrációs tesztek írását és futtatását valamely keretrendszer használatával (JUnit, TestNG, stb.).	Ismer legalább egy unit és integrációs teszt eszközt (JUnit, TestNG, stb.), ismeri azok képességeit, funkcióit.	Elkötelezett abban, hogy tesztekkel lefedje az általa készített kódot a használt teszteszköz funkcióinak kihasználásával.	Önállóan teszt eszközt választ és teszteseteket ír, melyekkel a funkcionális helyesség biztosítható és automatizáltan újra tesztelhető.
Megtanítandó és elsajátítandó témakör(ök); kapcsolódó foglalkozás(ok) megnevezése, (óra)száma és tartalmi elemei:				
Témaköre neve: Unit és integrációs tesztelés Spring Boot környezetben (40 óra) Foglalkozás: Unit és integrációs tesztelés Spring Boot környezetben (40 óra) Tartalmi elemei: <ul style="list-style-type: none"> • Unit tesztelés • Integrációs tesztelés • RestTemplate 				
S.sz.:	Készségek, képességek:	Ismeretek:	Elvárt viselkedésmódok, attitűdök:	Önállóság és felelősség mértéke:
4.	Képes felépíteni egy háromrétegű alkalmazást valamilyen keretrendszer használatával (pl. Spring Framework, Spring Boot, Java EE, valamilyen MicroProfile implementáció, tetszőleges microservice keretrendszer, stb.)	Ismer egy háromrétegű alkalmazások felépítésére használható keretrendszert, annak Dependency Injection képességeit.	Törekszik arra, hogy az alkalmazásait három réteggel építse fel (prezentációs, üzleti logika, perzisztens). Erre valamilyen keretrendszert használ, ami támogatja a rétegek megvalósítását, azok összekapcsolását.	Önállóan épít össze egy háromrétegű alkalmazást, és a komponenseket a megfelelő rétegekben helyezi el.
Megtanítandó és elsajátítandó témakör(ök); kapcsolódó foglalkozás(ok) megnevezése, (óra)száma és tartalmi elemei:				
Témaköre neve: Alkalmazás architektúra (10 óra) Foglalkozás: Alkalmazás architektúra (10 óra)				

	Tartalmi elemei:			
	<ul style="list-style-type: none"> Alkalmazás architektúra Három rétegű alkalmazás létrehozása 			
S.sz.:	Készségek, képességek:	Ismeretek:	Elvárt viselkedésmódok, attitűdök:	Önállóság és felelősség mértéke:
5.	Képes felépíteni egy tisztán objektumorientált, clean code szabályokat alkalmazó üzleti logika réteget. Ezt megfelelően leválasztja a prezentációs és perzisztens rétegtől.	Ismeri azokat a technikákat, hogy tud üzleti entitásokat és folyamatokat modellezni és implementálni.	Törekszik arra, hogy egy könnyen olvasható és karbantartható, a többi rétegtől független üzleti réteget alakítson ki.	Önállóan alakít ki olyan üzleti logika réteget, mely tartalmazza az üzleti elvárások megvalósítását, az üzleti entitásokat és folyamatokat.
Megtanítandó és elsajátítandó témakör(ök); kapcsolódó foglalkozás(ok) megnevezése, (óra)száma és tartalmi elemei:				
Témaköre neve: Üzleti logika (10 óra) Foglalkozás: Üzleti logika (10 óra) Tartalmi elemei: <ul style="list-style-type: none"> Üzleti logika Clean Code alapelvek Üzleti logika implementációja 				
S.sz.:	Készségek, képességek:	Ismeretek:	Elvárt viselkedésmódok, attitűdök:	Önállóság és felelősség mértéke:
6.	Képes rétegenként külön teszteseteket implementálni valamilyen teszt keretrendszer használatával (JUnit, TestNG, Spring Test, Arquillian, stb.)	Ismer egy eszközt, melynek segítségével kevesebb kóddal képes rétegenkénti tesztelést felépíteni.	Törekszik arra, hogy az alkalmazását rétegenként tesztelje. A külön letesztelt rétegek összeépítése után már kisebb a hibalehetőség.	Önállóan ír teszteseteket úgy, hogy csak egy adott réteget tesztel vele.
Megtanítandó és elsajátítandó témakör(ök); kapcsolódó foglalkozás(ok) megnevezése, (óra)száma és tartalmi elemei:				
Témaköre neve: Integrációs tesztelés (10 óra) Foglalkozás: Integrációs tesztelés (10 óra) Tartalmi elemei: <ul style="list-style-type: none"> Üzleti logika réteg tesztelése Controller réteg tesztelése Repository réteg tesztelése 				

4.3. Tananyagegység

4.3.1.	Megnevezése:	Adatbáziselérés JPA és Spring Data JPA használatával
4.3.2.	Célja:	A tananyagegység célja az adatbáziskezelés és ORM bemutatása. Hogyan kell fejlett eszközökkel adatbázisba írni, és onnan olvasni.
4.3.3.	Megvalósítása során alkalmazott munkaformák:	Frontális oktatás, mentor által felügyelt egyéni- és csoportmunka, távoktatás
4.3.4.	Megvalósítása	Előadás, magyarázat, szemléltetés, gyakorlat, megbeszélés, együttes és önálló



	során alkalmazott képzési módszerek:	tananyag feldolgozás, önálló tanulás, csoportos feladatmegoldás		
4.3.5.	Óraszám:	80 óra		
4.3.6.	Beszámítható óraszám ³ :	80 óra		
4.3.7.	A programkövetelményben meghatározott, a tananyagegység keretében elsajátítandó szakmai követelmények			
S.sz.:	Készségek, képességek:	Ismeretek:	Elvárt viselkedésmódok, attitűdök:	Önállóság és felelősség mértéke:
1.	Az adatokat relációs adatbázisban tárolja és azokhoz hozzáférést biztosít. Ehhez valamilyen ORM technológiát használ (JPA, Hibernate, EclipseLink, stb.), mely automatikusan leképezi az objektumokat táblákra. Ennek használatával alakítja ki a perzisztens réteget.	Ismer legalább egy olyan ORM technológiát, mellyel SQL használata nélkül automatikusan tud objektumokat leképezni relációs adatbázisba.	Törekszik arra, hogy a megfelelő adatok tárolása és betöltése ORM eszközzel történjen.	Önállóan alakít ki adatbázis sémát, majd programból ahhoz kapcsolódik és adatokat ment, módosít, felolvas és töröl valamely ORM eszköz használatával.
Megtanítandó és elsajátítandó témakör(ök); kapcsolódó foglalkozás(ok) megnevezése, (óra)száma és tartalmi elemei:				
Témaköre: JPA (50 óra) Foglalkozás neve: JPA (50 óra) Tartalmi elemei: <ul style="list-style-type: none"> • Mentés és lekérdezés • Integrációs tesztelés • Entitások és életciklusuk • Kulcsok • Lekérdezések 				
S.sz.:	Készségek, képességek:	Ismeretek:	Elvárt viselkedésmódok, attitűdök:	Önállóság és felelősség mértéke:
2.	Inicializációs eszközt használ az adatbázis sémájának létrehozására (pl. Flyway, Liquibase, stb.).	Ismer egy séma inicializációs eszközt, mely létrehozza az adatbázisban a szükséges objektumokat (táblákat, view-kat, tárolt eljárásokat, stb.).	Szem előtt tartja, hogy az alkalmazás maga is képes legyen létrehozni és migrálni az adatbázis sémát.	Önállóan adatbázis sémát inicializáló keretrendszert használ.
Megtanítandó és elsajátítandó témakör(ök); kapcsolódó foglalkozás(ok) megnevezése, (óra)száma és tartalmi elemei:				
Témaköre: Séma migráció (10 óra) Foglalkozás neve: Séma migráció (10 óra) Tartalmi elemei: <ul style="list-style-type: none"> • Séma migráció Flyway vagy Liquibase eszközökkel 				
S.sz.:	Készségek, képességek:	Ismeretek:	Elvárt viselkedésmódok, attitűdök:	Önállóság és felelősség mértéke:

³ Kontaktórától eltérő munkaforma alkalmazása esetén, ha az a képzés óraszámába beszámítható

2.	Objektumok közötti kapcsolatokat is képes perzisztálni.	Ismeri, hogy hogy lehet több, egymással kapcsolatban álló entitást adatbázisba menteni. Ismeri a felolvasási stratégiákat (eager, lazy betöltés).	Törekszik arra, hogy a kapcsolatok betöltése hatékonyan történjen, csak az legyen betöltve, melyre szükség van.	Önállóan dönt a tárolás és betöltés módjáról kapcsolódó entitások esetén.
	Megtanítandó és elsajátítandó témakör(ök); kapcsolódó foglalkozás(ok) megnevezése, (óra)száma és tartalmi elemei:			
	Témaköre: JPA kapcsolatok kezelése (10 óra) Foglalkozás neve: JPA kapcsolatok kezelése (10 óra) Tartalmi elemei: <ul style="list-style-type: none"> • Többértékű attribútumok • Kapcsolatok 			
S.sz.:	Készségek, képességek:	Ismeretek:	Elvárt viselkedésmódok, attitűdök:	Önállóság és felelősség mértéke:
3.	Képes tranzakciókezelést használni (programozott vagy deklaratív módon).	Ismeri az adott keretrendszer tranzakciókezelési megoldásait (programozott vagy deklaratív), a kapcsolódó metódusokat és annotációkat.	Törekszik arra, hogy hatékony tranzakciókezelést használjon, ne nyújtsa feleslegesen hosszúra a tranzakciókat.	Önállóan dönt a tranzakciós határok meghúzásáról.
	Megtanítandó és elsajátítandó témakör(ök); kapcsolódó foglalkozás(ok) megnevezése, (óra)száma és tartalmi elemei:			
	Témaköre: Tranzakciókezelés (10 óra) Foglalkozás neve: Tranzakciókezelés (10 óra) Tartalmi elemei: <ul style="list-style-type: none"> • Deklaratív tranzakciókezelés 			

4.4. Tananyagegység

4.4.1.	Megnevezése:	Projektmunka
4.4.2.	Célja:	A projektmunka célja egy olyan komplex alkalmazás elkészítése, melyben szerepelnek az eddig tanult technológiák.
4.4.3.	Megvalósítása során alkalmazott munkaformák:	Frontális oktatás, mentor által felügyelt egyéni- és csoportmunka, távoktatás
4.4.4.	Megvalósítása során alkalmazott képzési módszerek:	Előadás, magyarázat, szemléltetés, gyakorlat, megbeszélés, együttes és önálló tananyag feldolgozás, önálló tanulás, csoportos feladatmegoldás
4.4.5.	Óraszám:	80 óra
4.4.6.	Beszámítható óraszám ⁴ :	80 óra

⁴ Kontaktórától eltérő munkaforma alkalmazása esetén, ha az a képzés óraszámába beszámítható

4.4.7. A programkövetelményben meghatározott, a tananyagegység keretében elsajátítandó szakmai követelmények				
S.sz.:	Készségek, képességek:	Ismeretek:	Elvárt viselkedésmódok, attitűdök:	Önállóság és felelősség mértéke:
	Komplex alkalmazás készítése	Ismeri, hogyan kell az eddig tanult technológiákat gyakorlatban alkalmazni egy komplex alkalmazás megvalósításához.	Törekszik arra, hogy az általa megírt alkalmazás megvalósítsa a funkcionális követelményeket.	Önállóan képes megvalósítani egy komplex alkalmazást a tanult technológiák használatával.
Megtanítandó és elsajátítandó témakör(ök); kapcsolódó foglalkozás(ok) megnevezése, (óra)száma és tartalmi elemei:				
1.	<p>Témaköre: Projektmunka (80 óra) Foglalkozás neve: Projektmunka (80 óra) Tartalmi elemei:</p> <ul style="list-style-type: none"> ● Specifikáció megismerése ● Tervek elkészítése ● Osztályok elkészítése ● Adatbáziskezelés ● REST API elkészítése ● Séma migráció ● Unit és integrációs tesztek ● A projektmunka során JDK-t, Maven, IntelliJ IDEA Community fejlesztőeszközt kell használnia. A projektet Git verziókezelőben kell beadnia. A projekthez JUnit teszteseteket kell írnia. 			

5. Csoportlétszám

5.1.	Maximális csoportlétszám:	80 fő (Zárt rendszerű elektronikus távoktatás esetén nem releváns.)
------	---------------------------	--

6. A képzésben részt vevő teljesítményét értékelő rendszer leírása

Szakképzés esetén: (Szkt. végrehajtásáról szóló 12/2020. (II. 7.) Korm. rendelet 20. § (1) bekezdés b) pont.) A szakmai képzés képzési programja tartalmazza a képzésben részt vevő személy tanulmányi munkájának írásban, szóban vagy gyakorlatban történő ellenőrzési és értékelési módjait, diagnosztikus, szummatív, fejlesztő formáit is.

Előzetes tudásmérés (diagnosztikus értékelés):	
Előzetes tudásmérés: annak felmérése, hogy a képzésre jelentkező dokumentumokkal nem igazolt tanulmányai vagy megszerzett gyakorlati tapasztalatai alapján képes-e a képzés során elsajátítandó tananyagegység követelményeinek teljesítésére, amelynek eredményeként a követelmények megfelelő szintű teljesítése esetén a tananyagegység elsajátítására irányuló képzési rész alól a képzésre jelentkezőt fel kell menteni.	
6.1.	<p>A képzésre jelentkező kérésére előzetes tudásmérést biztosítunk.</p> <p>Az előzetes tudásmérés az adott tananyagegységben megszerezhető kompetenciákra terjed ki, melyet szóbeli kérdések és/vagy gyakorlati feladatok megoldásán keresztül mérünk fel képzési részenként.</p> <p>Megszerezhető minősítések: „Megfelelt” vagy „Nem felelt meg”</p> <p>„Megfelelt” minősítéshez tartozó követelményszint: Legalább 81%-ot elérő eredmény</p>

6.2.	Képzés közbeni (fejlesztő) értékelés: A fejlesztő értékelés szerepe, hogy a képzésben résztvevők fejlődését támogassa, a tanulási igényeket pontosítsa, az oktatók tanulásszervezési feladatait segítse. A képzés közbeni fejlesztő értékelés módjai lehetnek: Visszakérdezés, Gyakorlati feladatmegoldás, Képzésben résztvevő visszajelzései, Beszélgetés A fejlesztő értékeléshez nem tartozik minősítés , a tanulási és tanítási folyamatokat szolgálja. A képzés során 1 db félidős (rész) vizsgán kell megfelelnie a résztvevőnek. A részvizsga két részből áll: - A teszt 20 db feleletválasztásos tesztkérdést tartalmaz. A feleletválasztásos tesztkérdéseket úgy kell kialakítani, hogy egyetlen helyes válaszlehetőség legyen lehetséges. A vizsgatevékenység végrehajtására rendelkezésre álló időtartam 45 perc. A helyes válasz 5 pontot ér, a helytelen válasz 0 pontot, így maximálisan 100 pontot lehet elérni. A sikeres vizsgához 50 pont megszerzése szükséges. Sikertelen részvizsga esetén a vizsga megismételhető - Gyakorlati vizsga (összesen 80 pont), 5 részfeladatból áll, pontozása a következő: - 0 - 5 pont: az adott projekt nem fordul le, többségében nem futnak le a tesztesetek, az elvárt funkcionálitást nem valósítja meg. Azonban több teszteset nem fut le, és a kód is olvashatatlan. - 6 - 11 pont: a projekt lefordul, a tesztesetek legtöbbje lefut, ezek nagyrészt funkcionálisan is helyesek, és a clean code elvek nagyrészt betartásra kerültek. - 12 - 16 pont: ha a projekt lefordul, a tesztesetek lefutnak, funkcionálisan helyesek, és csak apróbb funkcionális vagy clean code hibák szerepelnek a megoldásban. A vizsgatevékenység akkor eredményes, ha a vizsgázó a megszerezhető összes pontszám legalább 51 %-át elérte. Sikertelen részvizsga esetén a vizsga (2 alkalommal) megismételhető.
	Részvevő záró (szummatív) értékelése: A képzés záróvizsgával zárul. A záróvizsga a képzés végén kerül megtartásra, és két részből áll. Tartalmaz egy írásbeli részt, és egy projektfeladatot. Az írásbeli rész tesztkérdéssor 20 kérdésből áll. A feleletválasztásos tesztkérdéseket úgy kell kialakítani, hogy egyetlen helyes válaszlehetőség legyen lehetséges. A vizsgatevékenység végrehajtására rendelkezésre álló időtartam 45 perc. A helyes válasz 5 pontot ér, a helytelen válasz 0 pontot, így maximálisan 100 pontot lehet elérni. A tesztkérdések a következő témakörök ismeretét mérik fel: <ul style="list-style-type: none">● Unit és integrációs tesztelés elméleti alapjai (4 kérdés)● Többrétegű alkalmazás és dependency injection (4 kérdés)● API tervezés és implementáció (4 kérdés)● ORM eszközök, perzisztens réteg használatának elméleti alapjai (4 kérdés)● Konténerizáció (4 kérdés)● A projektfeladat programozási feladatból áll. A programozási feladatban egy többrétegű backend alkalmazást kell elkészítenie, melyeknek szöveges leírásnak és előre átadott teszteseteknek kell megfelelnie. A feladat az alábbi fejlesztési részfeladatokat kell tartalmazza:
6.3.	

	<ul style="list-style-type: none">• prezentációs réteg, API kialakítása• üzleti logika réteg kialakítása• üzleti entitások kialakítása• perzisztens réteg kialakítása• konténerizáció megvalósítása <p>A vizsgázó a feladatot szöveges formátumban kapja meg. A feladatléírás tartalmazza a feladat szöveges leírását. A projekt vázát, valamint az automata teszteseteket verziókövető rendszeren keresztül kapja meg. A vizsgázó feladata a szöveges leírás megértése, a tesztesetek értelmezése. Ezek alapján meg kell terveznie a megoldást. A tervezés során meg kell határozni a megoldást biztosító komponenseket.</p> <p>A tervezés után implementálnia kell a megoldást a megfelelő fejlesztőeszközben:</p> <ul style="list-style-type: none">• Meg kell nyitnia a projekt vázát, melyben a tesztesetek nem fognak lefordulni.• A vizsgázónak létre kell hoznia a megtervezett komponenseket, mely után a projekt lefordítható, de funkcionálisan nem működőképes, azaz a tesztesetek már lefutnak, de hibát adnak.• Ezután úgy kell implementálnia a hiányzó részeket, hogy mind a leírásnak, mind a clean code alapelveknek, mind a teszteseteknek megfeleljenek, azaz a tesztesetek lefuttatása sikeres legyen.• Az alkalmazást le kell buildelni. <p>A munkája során figyelnie kell a területen szokásos paradigmák és elvek betartására, és olvasható és karbantartható kódot kell írnia. Amennyiben a vizsgázó elkészült a munkájával, azt verziókövető rendszeren, webes felületen keresztül vagy e-mailen kell beadnia.</p> <p>A vizsgafeladat elkészítésére 195 perc áll a vizsgázó rendelkezésére.</p> <p>A záróvizsga összesen 255 perc, 45 perc a tesztkérdéssorozat, és 210 perc a projektfeladat. A záróvizsgán összesen 200 pont szerezhető, 100 pont a tesztkérdésekkel, és 100 pont a projektfeladattal.</p> <p>A záróvizsgán megszerezhető minősítések és a megszerezhető minősítésekhez tartozó követelményszintek:</p> <ul style="list-style-type: none">• Megfelelt: a vizsgatevékenység során a képzésben résztvevő legalább 100 pontot elért.• Nem felelt meg: a vizsgatevékenység során a képzésben résztvevő 99 vagy annál kevesebb pontot ért el. <p>Sikertelen záróvizsga esetén a záróvizsga megismételhető.</p>
--	---

7. A képzés, a képzés egyes tananyagegységeinek elvégzéséről szóló igazolás kiadásának feltételei

7.1.	A képzés elvégzéséről szóló igazolás megnevezése:	TANÚSÍTVÁNY 2013. évi LXXVII. törvény 13/B. § 11/2020. (II. 7.) Korm. rendelet 22. § (1)
7.2.	A képzés elvégzéséről szóló igazolás kiadásának feltételei:	A képzés elvégzéséről szóló tanúsítvány kiadásának feltétele a záróvizsgán „Megfelelt” minősítés megszerzése.
7.3.	A képzés egyes tananyagegységeinek elvégzéséről szóló igazolás kiadásának feltételei:	A tananyagegységek elvégzéséről nem kerül kiadásra külön igazolás.

8. A képzési program végrehajtásához szükséges feltételek

8.1.	Személyi feltételek:	Felsőfokú iskolai végzettség és legalább 2 év a képzési program tananyagához illeszkedő szakirányú szakmai tapasztalat, vagy középfokú iskolai végzettség és legalább 3 év a képzési program tananyagához illeszkedő szakirányú szakmai tapasztalat.
8.2.	Személyi feltételek biztosításának módja:	Az oktatót képző intézmény foglalkoztatja munkaszerződéssel, megbízási szerződéssel vagy az oktató alkalmazását bizonyító más szerződéssel.
8.3.	Tárgyi feltételek:	Kontaktórák képzési rész esetén résztvevői létszámnak megfelelő: <ul style="list-style-type: none"> - asztal, szék - számítógép (javasolt: Core i5, 8GB RAM, 40GB háttértár), internetkapcsolat Nem kontaktórák képzési rész esetén: <ul style="list-style-type: none"> - személyi számítógép vagy laptop, lokális rendszergazdai (adminisztrátori) jogosultsággal (javasolt: minimum Core i5 vagy ennek megfelelő AMD processzor, minimum 8 GB RAM és legalább 40 GB szabad tárhely), - Microsoft Windows 10 vagy Linux operációs rendszer, - mikrofon (beépített, vagy külső) - javasolt webkamera, valamint szélessávú (legalább 5 Mbit/sec szabad sávszélességű) internetelérés.
8.4.	Tárgyi feltételek biztosításának módja:	Kontaktórák képzési rész esetén: a képzési helyszínek, oktatótermek meglétét felnőttképző tulajdonjog, használati jog vagy bérleti jogviszony alapján biztosítja; a tárgyi eszközök meglétét képzésben résztvevő saját eszközeként, vagy a felnőttképző tulajdonjog, használati jog vagy bérleti jogviszony alapján biztosítja. Nem kontaktórák képzési rész esetén a tárgyi eszközök meglétét képzésben résztvevő saját eszközeként biztosítja.
8.5.	A képzéshez kapcsolódó egyéb speciális feltételek:	-
8.6.	A képzéshez kapcsolódó egyéb speciális feltételek biztosításának módja:	-

9. Képesítő vizsga

A képesítő vizsgát nem a képző intézmény szervezi és bonyolítja.

A szakmai képzéshez kapcsolódóan megszerezhető szakképesítés megszerzésére irányuló képesítő vizsgát a nemzeti akkreditálásról szóló törvény szerinti akkreditáló szerv által személytanúsító szervezetként **akkreditált vizsgaközpont szervezhet.**

A szakmai képzéshez kapcsolódóan megszerzett képesítő bizonyítvány államilag elismert, önálló végzettségi szintet nem biztosító szakképesítést tanúsít.

A képesítő vizsga megszervezéséhez szükséges feltételek és a képesítő vizsga vizsgatevékenységeinek részletes leírása a <https://szakkesites.ikk.hu/> weblapon érhető el a programkövetelmények menüpont alatt.


A képesítő vizsgára bocsátás feltétele:

A szakmai képzés követelményeinek teljesítéséről (7.1. pont) a képző intézmény által a felnőttképzési adatszolgáltatási rendszerben kiállított tanúsítvány.

Egyéb feltételek: -----



10. Az előzetes minősítés ténye

Szakértő nyilatkozata:	A képzési program előzetes minősítése megtörtént.
Az előzetes minősítés helye:	Budapest
Az előzetes minősítés időpontja:	2021.06.04.
Az előzetes minősítést végző felnőttképzési szakértő neve:	Pádár Tivadar
Az előzetes minősítést végző felnőttképzési szakértő nyilvántartási száma:	FSZ/2020/000057
Felnőttképzési szakértő aláírása: (papíralapú képzési program esetében)	
Felnőttképző képviselőjére jogosult személy aláírása:	