A 06135011 számú Junior vállalati Java backend fejlesztő megnevezésű szakképesítés megszerzésére irányuló szakmai képzéseket megalapozó programkövetelmény

1. A programkövetelmény, illetve az ennek 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

2. A programkövetelmény alapján szervezhető szakmai képzéssel megszerezhető szakképesítés

2.1. Megnevezése: Junior vállalati Java backend fejlesztő

2.2. Szintjének besorolása

2.2.1. Az Európai Képesítési Keretrendszer (EKKR) szerint: 5

2.2.2. A Magyar Képesítési Keretrendszer (MKKR) szerint: 5

2.2.3. A Digitális Kompetencia Keretrendszer szerint: 6

3. A programkövetelmény alapján szervezhető szakmai képzéssel megszerezhető szakképesítés és az azzal betölthető munkakör vagy végezhető tevékenység kapcsolata, összefüggése:

3.1. 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.

3.2. A szakmai képzéshez kapcsolódóan megszerezhető szakképesítés jogszabályban meghatározott képesítési követelmény munkakör betöltéséhez vagy tevékenység folytatásához.

A képesítési követelményt előíró jogszabály:

4. A programkövetelmény alapján szervezhető szakmai képzéshez kapcsolódóan megszerezhető szakképesítéssel ellátható legjellemzőbb munkaterület, tevékenység vagy munkakör leírása:

A Junior vállalati Java backend fejlesztő képes egy alkalmazás többrétegű backendjét felépíteni, mely más alkalmazások által meghívható API-val rendelkezik. Olyan nagyvállalati keretrendszereket és technológiákat használ, melyek kielégítik a sokszor csak

nagyvállalatoknál jelentkező igényeket. Ilyen szabvány pl. a Java Enterprise Edition és ennek implementációi, vagy ezzel kompetitív Spring Framework/Spring Boot keretrendszer, stb. Ismer olyan ORM technológiát, mellyel adatbázishoz képes hozzáférni, és tud programozott vagy deklaratív tranzakciókezelést használni. El tudja készíteni az alkalmazás üzleti logika rétegét, használva a vállalati backend technológia különböző lehetőségeit (Dependency Injection, Clean Code). Ezen túl az alkalmazáshoz szabványokon alapuló API-t képes fejleszteni. Az alkalmazás komponenseihez unit és integrációs teszteket tud írni és futtatni. A fejlesztés során konténerizációs technológiát tud használni.

5. A programkövetelmény alapján szervezhető szakmai képzéssel megszerezhető szakképesítéshez szükséges képzési tartalom szabadalmi vagy szerzői jogi oltalom alatti állása:

5.1. Szabadalmi vagy szerzői jogi oltalom alatt áll: -

5.1.1. Az oltalom típusának megjelölése:

5.1.2. Nyilvántartó hatóság:

5.1.3. Azonosító vagy nyilvántartásba vételi száma:

6. A programkövetelmény alapján szervezhető szakmai képzés megkezdéséhez szükséges bemeneti feltételek:

6.1. Iskolai előképzettség: középfokú végzettség

6.2. Szakmai előképzettség: 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.

6.3. Egészségügyi alkalmassági követelmény: nem szükséges

6.4. Szakmai gyakorlat területe és időtartama: -

7. A programkövetelmény alapján szervezhető szakmai képzés elvégzéséhez szükséges foglalkozások minimális és maximális óraszáma (Amennyiben a programkövetelmény modulszerű felépítésű, a minimális óraszám a <u>modulonként</u> meghatározott minimális, a maximális óraszám a modulonként meghatározott maximális óraszámok összege):

7.1. Minimális óraszám: 160

7.2. Maximális óraszám: 320

8. A szakmai követelmények leírása:

8.1. Nem modulszerű felépítés esetén:

Képesség	Tudás	Attitűd	Autonómia és felelősség
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.
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.
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.
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.

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.
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 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.
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.
Alkalmazásokat és az adatbázist konténerizált környezetben futtat (pl. Docker).	Ismer egy konténerizációs technológiát és hogy hogyan kell konténereket létrehozni és kezelni. Ismeri annak módját, hogy	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.

	hogyan csomagolja az alkalmazást valamilyen konténerbe.		
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.
Az alkalmazáshoz, melyet fejleszt, unit és integrációs teszteket ír és futtat 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.
A teszteseteket képes különböző környezetekben futtatni (akár fejlesztőeszközben, mint IntelliJ IDEA, Eclipse, NetBeans, stb., akár build eszköz segítségével, mint Maven, Gradle, stb.)	Ismeri azokat a lehetőségeket, hogy hogyan lehet futtatni a unit és integrációs teszteket.	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.
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.

Unit tesztelésnél a kapcsolódó komponenseket mockolja erre használatos keretrendszerrel (pl. Mockito, PowerMock, stb.)	Ismer egy mock keretrendszert, mellyel a kapcsolódó komponenseket tudja kicserélni tesztelésre előkészített párjukkal (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 párjukkal.
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.

8.2. A szakmai képzés megszervezhető kizárólag távoktatásban: igen/nem

9. A programkövetelmény alapján szervezhető szakmai képzéssel megszerezhető szakképesítés társadalmi-gazdasági hasznosíthatóságának bemutatása (munkaerő-piaci relevanciája):

A Junior vállalati Java backend fejlesztő önállóan képes olyan backend alkalmazásokat fejleszteni, melynek van programozott interfésze (API), bonyolult üzleti logikát tartalmaz, és az adatokat képes adatbázisban tárolni. A jelenleg használt, böngészőből elérhető alkalmazások backend-jének nagy része ilyen, és egyre több igény mutatkozik ilyen alkalmazások fejlesztésére, vagy a már meglévő alkalmazások továbbfejlesztésére. Minden szektorban lehet ilyen alkalmazásokat találni, úgymint bankszektor, telekommunikáció, energetika, közigazgatás stb. A meglévő régi, un. legacy alkalmazások kiváltása, webes alkalmazásra történő lecserélése is folyamatos feladat, ami állandó munkát biztosít a Junior vállalati Java backend fejlesztőnek.

10. A képesítő vizsga megszervezéséhez szükséges feltételek és a képesítő vizsga vizsgatevékenységeinek részletes leírása:

10.1. A képesítő vizsgára bocsátás feltétele:

A szakmai képzés követelményeinek igazolásáról a képző intézmény által, a felnőttképzési adatszolgáltatási rendszerben kiállított tanúsítvány.

Egyéb feltételek: -

10.2. Írásbeli vizsga

10.2.1. A vizsgatevékenység megnevezése: Junior vállalati Java backend fejlesztő írásbeli vizsga

10.2.2. A vizsgatevékenység, vagy részeinek leírása:

A vizsgatevékenység egy feleleletválasztós teszt, amelynek célja a képzés elméleti tudásanyagának számonkérése. Az írásbeli vizsga kérdéseit a következők szerint kell összeállítani:

- Kérdések száma, típusa: 20 db feleletválasztásos tesztkérdés, kérdésenként legalább 4, legfeljebb 6 válaszlehetőséggel, amelyből egyetlen a helyes válasz.
- A tesztkérdések témakörei és témakörönkénti darabszámai a következők kell legyenek:
    - 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)

10.2.3. A vizsgatevékenység végrehajtására rendelkezésre álló időtartam: 45 perc

10.2.4. A vizsgatevékenység aránya a teljes képesítő vizsgán belül: 20%

10.2.5. A vizsgatevékenység értékelésének szempontjai:

Az írásbeli vizsgát a következők szerint kell értékelni:

Maximálisan elérhető pontszám/százalék: 100 pont/100%

- 20 x 5 pont = 100 pont/100%.

Egyéb értékelési szempontok az írásbeli vizsgaértékeléssel kapcsolatban:

- A helyes válasz 5 pontot ér, a helytelen válasz 0 pontot ér.
- A rossz válasz megjelöléséért pontlevonás nem jár.

10.2.6. [1]A vizsgatevékenység akkor eredményes, ha a vizsgázó a megszerezhető összes pontszám legalább 51%-át elérte. Törtpontszámos eredmény esetén a kerekítés szabályait szükséges alkalmazni.

10.3. Projektfeladat

[1] Módosítva: 2023.02.10.

10.3.1. A vizsgatevékenység megnevezése: Junior vállalati Java backend fejlesztő gyakorlati vizsga

10.3.2. A vizsgatevékenység, vagy részeinek leírása:

A gyakorlati vizsga egy vizsgaremekből és egy programozási feladatból áll.

A) Vizsgaremek feladatrész

A vizsgázónak a vizsgát megelőzően egy komplex backend alkalmazást kell lefejlesztenie saját döntése alapján egyénileg választott témában.

Az alkalmazásnak az alábbi elvárásoknak kell megfelelni:

- Életszerű, valódi problémára nyújt megoldást.
- Adattárolási és -kezelési funkciókat is megvalósít.
- Legalább három rétegből épül fel.
- Szabványoknak megfelelő API-t tartalmaz megfelelő hibakezeléssel (pl. OpenAPI szabvány alapú REST webszolgáltatás).
- A forráskódnak a tiszta kód elveinek megfelelően kell készülnie.

A vizsgaremek benyújtásának módja:

A kész csomagot a vizsga előtt minimum 7 nappal kell a vizsgabizottsághoz benyújtani verziókövető rendszeren keresztül (pl. GitHub, GitLab, BitBucket, stb.). A megosztott anyagnak tartalmaznia kell az alábbiakat:

- A szoftver forráskódja
- Az automata tesztesetek
- Az adatbázis sémát előállító szkriptek
- Konténert előállító szkript

A vizsgafeladat során a vizsgázó gyakorlati bemutatóval összekapcsolt szóbeli előadás formájában mutatja be a

- szoftver célját
- műszaki megvalósítását
- működését
- forráskódját

A vizsgaremek bemutatására és megvédésére maximum 15 perc áll a vizsgázó rendelkezésére.

B) Programozási feladat feladatrész

Egy programozási feladat, egy többrétegű backend alkalmazás, melyeknek szöveges leírásnak és előre átadott teszteseteknek kell megfelelnie. A feladat az alábbi fejlesztési részfeladatokat kell tartalmazza:

- prezentációs réteg, API kialakítása
- üzleti logika réteg kialakítása
- üzleti entitások kialakítása
- perzisztens réteg kialakítása
- konténerizáció megvalósítása

A vizsgázó a feladatot szöveges formátumban kapja meg. A feladatleírás 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 határoznia a megoldást biztosító komponenseket.

A tervezés után implementálnia kell a megoldást a megfelelő fejlesztőeszközben:

- 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 teszteseteknek megfeleljenek, azaz a tesztesetek lefuttatása sikeres legyen.
- Az alkalmazást le kell buildelni.

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.

A vizsgafeladat elkészítésére 195 perc áll a vizsgázó rendelkezésére.

10.3.3. A vizsgatevékenység végrehajtására rendelkezésre álló időtartam: 210 perc

10.3.4. A vizsgatevékenység aránya a teljes képesítő vizsgán belül: 80%

10.3.5. A vizsgatevékenység értékelésének szempontjai:

A vizsgatevékenységgel 100 pontot lehet szerezni, az alábbi bontásban:

A) A vizsgaremek feladatrész

A feladatra 0 és 40 pont közötti pont adható:

- az alkalmazás átfogó értékelése (a komplexitás és kidolgozottság mértéke, milyen mértékben és minőségben valósította meg a szoftver a kitűzött célt, az API használhatósága): 20 pont
- 0-5 pont: Adattárolási és -kezelési funkciókat is megvalósítása.
- 0-5 pont A rétegek megvalósítása, a rétegek a megfelelő funkcionalitással rendelkezzenek, és lazán kapcsolódjanak.
- 0-5 pont: A backend alkalmazáshoz fejlesztett API megvalósítása.
- 0-5 pont: A tiszta kód elveinek betartása.

B) Programozási feladat feladatrész

A feladatra 0 és 60 pont közötti pont adható:

- 0 – 15 pont: prezentációs réteg, API kialakítása (CRUD műveletek, validáció, hibakezelés)
- 0 – 5 pont: üzleti logika réteg (Clean Code elvek)
- 0 – 5 pont: üzleti entitások kialakítása (kapcsolatok reprezentálása, betöltés)
- 0 – 15 pont: perzisztens réteg (CRUD műveletek, kapcsolatok kezelése)
- 0 – 15 pont: teszteseteknek való megfelelés
- 0 – 5 pont: konténerizáció

A részletes, adott projektfeladatra személyre szabott értékelés a mérés-értékelési útmutatóban fog szerepelni.

10.3.6. [2]A vizsgatevékenység akkor eredményes, ha a vizsgázó a megszerezhető összes pontszám legalább 51 %-át elérte. Törtpontszámos eredmény esetén a kerekítés szabályait szükséges alkalmazni.

A vizsga akkor eredményes, ha a vizsgázó mind az írásbeli, mind a gyakorlati (projektfeladat/vizsgaremek) vizsgatevékenységet eredményesen teljesítette. Az eredményes vizsga esetén alkalmazandó érdemjegy ponthatárok (a %-os arány megegyezik az elért pontszámmal):

- 0-50%– elégtelen (1)
- 51-60% – elégséges (2)
- 61-70% - közepes (3)
- 71-80% - jó (4)
- 81%-100% - jeles (5)

[2] Módosítva: 2023.02.10.

10.4. A vizsgatevékenységek lebonyolításához szükséges személyi feltételek:

A vizsgabizottság legalább egy tagjának jártasnak kell lennie a vállalati Java backend technológiákban és legalább 2 éves szakmai gyakorlattal és/vagy felsőfokú szakirányú (informatikai) végzettséggel szükséges rendelkeznie.

A vizsga lebonyolításához szükséges technikai feltételek biztosítása/felügyelete és a vizsga zavartalan lebonyolítása érdekében egy technikai szakember (rendszergazda) biztosítása szükséges.

10.5. A vizsgatevékenységek lebonyolításához szükséges tárgyi feltételek:

- Számítógép / laptop
- Internetkapcsolat
- Szoftverek: verziókezelő kliens (pl. Git, fejlesztőeszköz), Java Development Kit, valamilyen fejlesztőeszköz (IntelliJ IDEA, Eclipse, NetBeans, stb.), egy relációs adatbázis (pl. MySQL, PostgreSQL, H2, stb.), API kliens (pl. Postman, fejlesztőeszköz, stb.), konténerizációs környezet (pl. Docker), a képzésen használt és a vizsgára nyilvánosságra hozott szoftverlista szerinti szoftverek

10.6. A vizsgatevékenységek alóli felmentések speciális esetei, módja, és feltételei: -

10.7. A képesítő vizsgán használható segédeszközökre és egyéb dokumentumokra vonatkozó részletes szabályok:

Papír és toll/ceruza használata megengedett.

A vizsgaközpont által ellenőrzött és jóváhagyott, a technikai feltételeknek megfelelő, saját számítógép használata engedélyezett.

A vizsgatevékenység végrehajtásához internetkapcsolat áll a vizsgázók rendelkezésére.

Az internetkapcsolat biztosításának módját és formáját az adott vizsgafeladathoz kiadott útmutató tartalmazza. Ennek megfelelően az internetkapcsolat korlátozódhat meghatározott internetes címekre és/vagy hozzáférési időtartamra, de mindenképpen biztosítani kell, hogy az internetkapcsolatot a vizsgázók kizárólag általános keresésre használhassák, mással történő kommunikációra vagy a vizsgához célirányosan elkészített anyagok letöltésére ne.

10.8. A vizsgatevékenységek megszervezésére, azok vizsgaidőpontjaira, a vizsgaidőszakokra vonatkozó sajátos feltételek:

[3]A képesítő vizsga online/virtuális formában is megszervezhető, a résztvevők biztonságos hitelesítésével (pl. képernyő-, élőkép, iratok bemutatásával és jegyzői hitelesítéssel).

11. A szakmai képzés megszervezéséhez kapcsolódó különös, egyedi, speciális feltételek

[3] Módosítva: 2023.02.10.