

Nemzetközi Érettségi VI.

Nagy Tibor

A számítástechnika tantárgy a Nemzetközi Érettségi programjában

A tantárgy nagy anyagot fog át, bemutatva a számítástechnika összes fontos témakörét. Jól hasznosítható alapot ad a tájékozódásra a mai gyorsan változó informatikai világban. Azok számára a leghasznosabb, akik ilyen irányban kívánnak tovább tanulni.

A Nemzetközi Érettségi program (International Baccalaureate, a továbbiakban IB) keretében a diákok 6 tantárgyat tanulnak.¹ Az első öt csoportból kötelező választani egy tantárgyat, a hatodik tantárgy vagy az első öt csoport valamelyikéből vagy a hatodik csoportból kerülhet ki. Három tantárgyat emelt szinten, három alapszinten kell tanulni. Az emelt szintű tananyag elsajátítására két év alatt összesen 240 óra, az alapszintű tantárgyakra 150 óra áll rendelkezésre.

A számítástechnika tantárgy egyrészt a Nemzetközi Érettségi tantárgyi szerkezete miatt került a hatodik csoportba, másrészt azért, mert igazán nem tartozik egyik más kategóriába sem. Vannak olyan elképzelések, amelyek a számítástechnikát a negyedik csoportba sorolnák át a matematikai mellé. Ezzel nem a tantárgy matematikai vonatkozásait szeretnék hangsúlyozni, inkább a számítástechnika növekvő fontosságát kívánják kifejezni. Néhány iskolában folyik egy kísérleti tantárgy kipróbálása is Information Technology in a Global Society (kb. Informatika a társadalomban) néven. Ez főleg az informatikai eszközök alkalmazásával és azok társadalmi, szociális hatásával foglalkozik.

A számítástechnika tantárgy a közelmúltban megújult, a korábbi Computing Studies váltotta fel. A változás mind a tananyagban, mind a módszerekben jelentős korszerűsítést hozott.

A tantárgy célja

A számítástechnika tantárgy keretében a következő készségeket és ismereteket sajátítják el a diákok:

- a számítástechnika alapvető fogalmainak megértése;
- hogyan történik problémák megfogalmazása és számítógépes megoldás megtervezése a gyakorlatban;
- algoritmusok tervezése és helyes alkalmazása;
- számítógépek és kapcsolódó informatikai eszközök felépítése;
- a számítástechnikai ismeretek segítségével új, elterjedőben levő technológiák megértése és használata;

¹ A tárgyak, amint korábbi írásainkban bemutattuk, a következő tantárgycsoportokból kerülnek ki: anyanyelvi irodalom, második idegen nyelv, társadalomtudományok, természettudományok, matematika, művészetek (képzőművészet, zene, klasszikus nyelvek (latin vagy görög), számítástechnika).

- számítógépek felhasználása adatfeldolgozásra, adatelemzésre, szimulációra, a kommunikáció hatékonyságának fejlesztésére;
- a számítástechnika alkalmazhatóságának határait és a vele való visszaélés lehetőségeit felismerve felelősséggel használni a számítógépet.

A tananyag bemutatása

Az IB-tanmenet készítésekor tekintettel voltak arra, hogy a tantárgyat alap- és emelt szinten is taníthatják ugyanabban az oktatási intézményben egy évfolyamon belül. Mivel az emelt szintű anyag az alapszintű ismeretek kiegészítése, lehetőség van arra, hogy az alap- és emelt szinten tanuló diákokat a két éves periódus egy részében együtt tanítsák. Ezzel az iskolai számítástechnikai kapacitás is jobban kihasználható.

Az alapszint tananyaga (zárójelben az előírányzott óraszámok)

1. A problémamegoldás módszertana (20 óra)

Hogyan kell megfogalmazni, mit szeretnék csinálni, olyan módon, hogy az tartalmazza: melyek a rendelkezésre álló adatok, ezekből milyen végeredményt szeretnék kapni, és ezt milyen módszerrel állítom elő. Módszerek algoritmusok tervezésére, az adatkezelés folyamatának megjelenítésére. 'Top-down design' (a feladat fokozatos széttagolása kisebb feladatokra), mint követendő problémamegoldó stratégia.

2. Számítógépprogramok készítése (50 óra)

Egy magasszintű nyelv alapjainak elsajátítása, összehasonlítás alacsony szintű nyelvvel. Az adatbevitel, az eredmény-megjelenítés eszközei, ciklus- és elágazásszervezés. Strukturált programozás, eljárások és függvények, különböző adattípusok használata. File-kezelés és file-szervezési módok. Rendezési és keresési algoritmusok. A programfejlesztés és hibakeresés eszközei. Tesztelés. Dokumentáció készítése.

3. Adatszerkezetek (20 óra)

Egyszerű és összetett adattípusok, logikai és halmazműveletek. A feladat megoldásához legjobban illeszkedő adattárolási mód megválasztása.

4. A számítástechnika matematikai eszközei (15 óra)

A számítástechnikában használatos számrendszerek (2-es, 8-as, 16-os számrendszer). Számábrázolás kettes számrendszerben, ennek pontossága és a számítások lehetséges hibái. Az adattárolás és átvitel helyességének ellenőrzése. Egyszerű logikai áramkörök tervezése és igazságtáblázatok.

5. Számítógép architektúrák (10 óra)

Számítógép-generációk. Főbb komponensek és perifériák. Az adatbevitel és -megjelenítés eszközei. Számítógép-hálózatok, adatátvitel és adatbiztonság. Operációs rendszerek. Adatbázis kezelés.

6. Alkalmazói programok használata (15 óra)

Könyvtári adatbázis, filekezelők, víruskeresők stb. Példák a legváltozatosabb területekről problémák megoldására (például irodai automatizálás, robotika, gyártás és folyamatvezérlés, mesterséges intelligencia, oktatás stb.)

Az emelt szint további témakörei (további 90 óra).

Számítástechnikai rendszerek tervezése. Objektum-orientált programozás. Összetett logikai formulák kiértékelése. Dinamikus adatszerkezetek: listák, fák. Rekurrenzio. Adatáramlás mélyebb vizsgálata. Digitális-analóg jelátalakítás. Adatvédelem.

Korábban említettem, hogy a számítástechnika tantárgy egy másik tárgyat, a 'Computing Studies'-t váltotta fel. A változtatást épp az indokolta, hogy korábban a számítástechnika túlságosan statikus, elméleti volt. A megújított tanmenet irányelveiben hangsúlyozza a gyakorlati megközelítés fontosságát valamennyi témakör esetében. Így erre mind a tanítás, mind a számonkérés során törekedni kell.

Tanítási módszerek

A tantárgy jellegéből fakadóan a tananyag elméleti és gyakorlati ismereteket is tartalmaz. Bizonyos részek tanítási sorrendje kötött, mert feltételezi a szükséges ismeretek meglétét, mások viszont szabadon elhelyezhetők a tanmenet tetszés szerinti helyén. Az órák szervezésében az bizonyult hasznosnak, ha két elméleti és két számítógép melletti órát tartunk. Ez jól illeszkedik a tantárgyi kettősséghez, lehetővé teszi a számítógéppark hatékonyabb kihasználását és az előkészített ismeretek alapján történhet a gyakorlati feladatok megoldása.

A legnehezebb feladat a programozás tanítása. A diákok eltérő képességekkel érkeznek, vannak egészen kezdők és haladók is. Ráadásul a tanulók némelyike vonzódik a programíráshoz, mások idegenkednek tőle. Itt jut szerephez az algoritmi-kus tervezés, mert nem közvetlenül egy konkrét programnyelven, hanem absztrakt eszközök például folyamatábra segítségével fogalmazzuk meg a probléma megoldást. Ezt legtöbbször az egyébként „gyengén programozók” is kedvelik és segítséget nyújt számukra a programkód megírásához. Azokat pedig, akik legszívesebben azonnal püfölni kezdenék a billentyűzetet, rászoktatja az átgondolt, áttekinthető programkészítésre. Néhány összekuszálódott és hibás program megírása után mindenki belátja, hogy az összetettebb programok komoly tervezést igényelnek.

A módszeres programozás megtanítása a tantárgy egyik központi feladata. Fontos, hogy már a kezdet kezdetén, egyszerű programokon mutassuk be a tervezés eszközeit, még akkor is, ha ilyenkor fölöslegesnek tűnhet az efféle előkészítés. Ha ugyanis a módszeres programtervezésre, a ki- és bemenő adatok helyes megválasztására csak a bonyolultabb programok elkészítésekor akarjuk megtanítani a diákokat, akkor már késő. Ebben a fázisban ezeknek a készségeknek már meg kell lenniük, hogy segítségükkel a bonyolult feladatot már (esetleg korábban megoldott) egyszerűbb részekre bontsuk fel.

Nehézséget jelent, hogy nincs egységes tankönyv. Több kötet is szerepel ugyan az ajánlott könyvek listáján, de igazából egyikük sem tartalmaz minden ismeretet és némelyik formailag is nehezen követhető. Ezért az anyagot sok forrásból kell összeválogatni, esetleg egy témát érdemes több feldolgozásban is a tanulók rendelkezésére bocsátani, hogy a szükséges ismereteket és szókincset jobban elsajátíthassák. Írásom végén közlöm azoknak a könyveknek a listáját, amelyeket a tanítás során hasznosnak találtam.

Érdemes megragadni minden lehetőséget, amivel a mindennapi élethez lehet kapcsolni a tananyagot. Ha a diákok külföldi csereprogramban vesznek részt, a tanulóknak kis tanulmányt kell készíteniük a meglátogatott intézmény számítástechnikai felszereltségéről, az alkalmazott programokról, a felhasználók köréről, egy esetleges könyvtári nyilvántartó rendszerről stb.

A különböző számrendszerek tanításakor feladatul lehet kitűzni olyan program elkészítését, amely elvégzi a számok átváltását egyik számrendszerből a másikba.

Az év közben beadandó programok témáját érdemes úgy megválasztani, hogy az elkészült program a diák számára hasznos legyen. Például egy történelmi

eseményeket, évszámokat nyilvántartó adatbázis nagyon hasznos lehet az érettségire való felkészüléskor.

Érdeemes megszervezni egy-egy kiállítás megtekintését is, ahol a tanulóknak alkalmuk lehet olyan eszközök és alkalmazások megtekintésére, amelyek az iskolában nem hozzáférhetők.

Az IB Computer Science tananyaga széles körű ismerteket ölel fel, ezek egy része az iskolában nem mutatható be. Egy egyetemi számítógépes központban vagy egy informatikai cégnél tett látogatás során lehetőség van nagy hálózatok és az Internet bemutatására, a tananyagban szereplő operációs rendszerek, adatbáziskezelő programok kipróbálására.

A dolgozatok elkészítésében elsősorban a korábban megírt érettségi dolgozatok jelentenek alapot. Szerepelnek még a programozási tudást ellenőrző kisebb feladatok és egy-egy témakört lezáró tesztek is.

Értékelés

Az IB többi tantárgyához hasonlóan az értékelés két részben történik. A külső értékelés az érettségi vizsga keretében központilag összeállított két dolgozat megírásából, a belső egy úgynevezett „Program Dossier” elkészítéséből áll.

Külső értékelés

A dolgozat kérdéseinek egy részére kötelező választ adni, mások pedig több kérdés közül választhatók.

A feladatok többsége gyakorlati jellegű, mint például: egy egyszerű algoritmus megírása, vagy egy kész algoritmus értelmezése, adott szempontok szerinti módosítása, vagy számábrázolással, logikai kifejezésekkel kapcsolatos számolás, vagy egyszerű file-kezelési feladat grafikus megoldása. Szerepelnek lexikális tudást ellenőrző kérdések, például:

- Neumann János szerepe a számítástechnikában,
- számítógéphálózatok fajtáinak jellemzése,
- egy strukturált programhoz kapcsolódó alapvető fogalmak,
- számítógép-vírusok jellemzői, az ellenük való védekezés módjai.

Egy tipikus vizsgakérdés az alábbi:

Egy kisvállalkozás menedzsere számítógépet vásárol. Te vagy az eladó, aki programcsomagokat is forgalmaz. Szeretnéd, ha a vásárló valamennyi programot megvásárolná a gépéhez. Írj két indokot az alábbi programok mindegyikéhez, amivel meggyőznéd őt arról, hogy szüksége lesz a használatukra: (a) Szövegszerkesztő, (b) Táblázatkezelő, (c) Adatbáziskezelő, (d) Levelezőprogram, (e) Kiadványszerkesztő.

Belső értékelés

A „Program Dossier” egy vagy legfeljebb három, a tanult magas szintű programozási nyelven elkészített és előírt szempontok szerint dokumentált programot tartalmaz. Ebben kell bemutatni, hogy a tanuló a gyakorlatban is elsajátította a programozás előírt elemeit és képes egy összefüggő, komolyabb program megírására.

A „dossier-program” témája tetszőleges. Egyetlen megkötés, hogy a programnak (programoknak) tartalmaznia kell a tanmenetben szereplő valamennyi programozási elemet és technikát. Használni kell egyszerű- és a felhasználó által

definiált adattípusokat, tömböket, rekordokat, eljárásokat és függvényeket. Szerepelnie kell file-kezelésnek, sorbarendezésnek vagy keresésnek. Az emelt szinten tanuló diák programjaiban ez kiegészül rekurzióval, mutató használattal, lista- és fa adatszerkezettel is.

Néhány példa „dossier-program” témaközből, ízelítőül: határidőnapló program, hanglemezkatalógus, egyszerű szövegszerkesztő, teniszjáték, életjáték, függvény-ábrázolás...

A tanmenet 20 iskolai órát szán a „dossier-program” fejlesztésére és további mintegy 30-50 órányi egyéni munkát feltételez a tanulótól. A tanár a programkészítés minden szintjén segítséget nyújt a diáknak, de a programot a tanulónak egyedül kell elkészítenie.

A „dossier” nem csupán a programlistából áll, a programlista igazából csak „függelék”. A „dossier” szerkezetileg hat részre tagozódik, ezek a következők:

1. A feladat megfogalmazása és a megoldás módjának leírása.

Meg kell mutatni, hogy a tanuló megértette a feladatot és alkalmas módszereket keresett a megoldáshoz. Itt kell szerepelnie a megoldás elméleti hátterének, és le kell írni a program célját, valamint alapvető funkcióit.

2. Algoritmikus reprezentáció

A tanult algoritmusleíró eszközök segítségével részfeladatokra bontva a kiindulási problémát, meg kell tervezni a programot, hogy az könnyen átültethető legyen egy magas szintű programozási nyelvre. Az algoritmusnak jól érthetőnek és áttekinthetőnek kell lennie.

3. Programlista

Az algoritmust a tanult programnyelvre (többnyire Pascalra) kell implementálni, a megvalósításban felhasználva az adott nyelv lehetőségeit. Az egyes változók és eljárások szerepét megjegyzésekben kell megmagyarázni és figyelni kell az áttekinthetőségre.

4. Tesztelés

A tanulónak meg kell győződnie arról, hogy nem követett el programozási hibát és programját minden előre látható helyzetre felkészítette. Ezt az adott program jellegétől függően változatos tesztadatok bevitelével, szélsőséges feladatokra való alkalmazásával kell kipróbálnia, és az eredményeket itt bemutatnia (például kinyomtatott képernyőképek formájában).

5. Felhasználói dokumentáció

A program lehetséges használóinak könnyen érthetően be kell mutatni, hogy a program hogyan és mire használható.

6. Értékelés

Az elkészült programról a diáknak véleményt kell alkotnia: mik a felhasználhatóság határai, és hogyan lehetne a programot továbbfejleszteni.

Az évközi iskolai értékelés dolgozatok írásából, beadandó programok és projektek elkészítéséből áll. A tanuló ezek és órai tevékenységének alapján kapja az iskolai érdemjegyeket.

Összegzés

A tantárgy nagy anyagot fog át, bemutatva a számítástechnika összes fontos témakörét. Jól hasznosítható alapot ad a tájékozódásra a mai gyorsan változó informatikai világban. Azok számára a leghasznosabb, akik ilyen irányban kívánnak tovább tanulni, mert rengeteg olyan elméleti és gyakorlati ismerethez juttatja

a diákokat, amelyeket később jól tudnak kamatoztatni. A tananyag megújulása segített abban, hogy a korábban túlságosan elméleti jellegű tantárgy közelebb kerüljön a mindennapok számítástechnikájához és vonzó választási lehetőséget kínáljon az érdeklődő diákok számára.

A Nemzeti alaptantervvel összehasonlítva kitűnik, hogy az IB tanmenet minden elemet tartalmaz a magyar tananyagból, azokat jelentősen kiegészíti és egy sor további ismeretet is magában foglal. Az IB jól tud támaszkodni a NAT tananyagára, különösen a programozási ismeretek tekintetében, így messzebb lehet eljutni a diákokkal a két év során. Ugyanakkor az IB eszközei és módszerei jól hasznosíthatók a normál tagozatok oktatásában.

Az IB-számítástechnika tanítása felkészültséget és sok munkát igényel, ugyanakkor kihívást, amelyben tanár és diák egyaránt próbára teheti képességeit. Ez egyike azoknak a tantárgyaknak, amelyek növekvő fontosságuk miatt valószínűleg egyre komolyabb szerepet játszanak majd az IB érettségi tantárgyai között.

Az IB Computer Science tantárgy tanítása során használt források

Computing: An Active Learning Approach *P. Heathcote*, DP Publications
A Glossary of Computing Terms Brit. Computer Soc., Cambridge University Press
Computer Science *C. S. French*, DP Publications
Understanding Computer Science *R. Bradley*, Stanley Thorne
Introduction to Pascal and Structured Design *Dale and Orshalick*, Heath and Co.
Computers and Applications *Daniel L. Slotnick*, Heath and Co.



Fekete Csilla (7. o.): A Szultán sátra Szigetváron 1566