

ՀՏԴ 37.02(004.45)

Ինֆորմատիկա

Վազգեն ԱՌՍՍԱՄՅԱՆ

ԱրՊՀ, մ.գ.թ., կիրառական մաթեմատիկայի և ինֆորմատիկայի ամբիոնի
դոցենտ

E-mail. varustamyan@rambler.ru

Արթուր ՄՆԱՅԱԿԱՆՅԱՆ

ԱրՊՀ, կիրառական մաթեմատիկայի և ինֆորմատիկայի ամբիոնի դասախոս

E-mail. mnacakanyan_artur77@mail.ru

ԳԻՏԵԼԻՔՆԵՐԻ ՍՏՈՒԳՄԱՆ ԵՎ ԳՆԱՀԱՏՄԱՆ ՀԱՄԱԿԱՐԳԻ ՄՇԱԿՈՒՄԸ

Սույն հոդվածում ընդհանրացվում են հեռավար ուսուցման ժամանակ գիտելիքի գնահատման մեթոդաբանական հիմնավորման առանձնահատկությունները, նշվում, որ ուսուցման գործընթացում պետք է հաշվի առնել ուսանողների կողմից խորը և հիմնավոր գիտելիքների յուրացումը ու հետադարձ կապի ապահովումը, որ գնահատումը նպատակաուղղված է սովորողների գիտելիքների, կարողությունների և հմտությունների ստուգման ու գնահատման առավել անաչառ, թափանցիկ ու արդյունավետ գործընթացի կազմակերպմանը:

Բանալի բառեր՝ Հեռավար ուսուցում, գիտելիքների գնահատում, ինքնուրույն աշխատանք, աշխատանքի արդյունավետություն, վերջնարդյունքների գնահատում, ուսանող, համադրելիություն, ավտոմատացում:

В. АРУСТАМЯН, А. МНАЦАКАНЯН
РАЗРАБОТКА СИСТЕМЫ ТЕСТИРОВАНИЯ И ОЦЕНКИ
ЗНАНИЙ

В статье дается краткое методологическое обоснование особенности оценивания знаний дистанционного обучения. При этом следует учесть приобретенные учащимися глубокие и

основательные знания и обеспечение взаимосвязи “учащийся-преподаватель”. Акцентируется внимание на том, что выставляемая оценка нацелена на проверку знаний, умений и навыков учащихся, и организацию более беспристрастного, прозрачного и эффективного процесса тестирования.

Ключевые слова: Дистанционное обучение, оценка знаний, самостоятельная работа, эффективность работы, оценка результатов, студент, сопоставимость, автоматизация.

***V.ARSTAMYAN, A.MNACAKANYAN
DEVELOPMENT OF KNOWLEDGE TESTING AND ASSESSMENT
SYSTEM***

This article summarizes the "methodological rationale" for distance learning assessment, noting that in-process learning should take into account students 'in-depth knowledge acquisition and feedback that assessment is aimed at learners' knowledge and skills. organizing a more impartial, transparent and effective skills testing and assessment process.

Key words: Distance learning, assessment of knowledge, independent work, work efficiency, evaluation of results, student, comparability, automation.

Հեռավար ուսուցումը, հանդիսանալով համակարգված ուսուցման ձև և ապահովելով փոխադարձ կապ սովորողի ու դասախոսի միջև, իրականացվում է տեղեկատվական տեխնոլոգիաների և հեռահաղորդակցության միջոցներով: Այն, ապահովելով կրթության մատչելիություն ու հասանելիություն, սովորողներին հնարավորություն է ընձեռում յուրացնելու հիմնական և լրացուցիչ մասնագիտական կրթական ծրագրեր չհեռանալով բնակության վայրից:

Հեռավար ուսուցման գործընթացը կազմակերպելով նոր տեխնոլոգիաներով և պահպանելով կրթության որակը՝ պետք է արագ հարմարվել տեխնոլոգիական փոփոխությունների ընթացքին:

ԱրՊՀ-ում գիտելիքների գնահատման բազմաբաղադրիչ համակարգն ապահովում է ուսանողների կողմից խորը և հիմնավոր գիտելիքների յուրացումը, դյուրացնում ուսումնառության և դրա գնահատման, հետադարձ կապի ապահովման գործընթացները, ինչպես նաև բարելավում ուսանողի կարգապահությունը և դասընթացներին մասնակցությունը: Ուսանողների ուսումնառության գնահատումը բարձրագույն կրթության ամենակարևոր

բաղադրատարրերից է, գիտելիքի գնահատման համակարգը պետք է ունենա գնահատման և նշագրման հստակ և հրապարակված չափանիշներ:

Բազմաթիվ գիտամանկավարժական ուսումնասիրություններ, ինչպես նաև մեր գործնական գործունեության փորձը ցույց են տալիս, որ ոչ բոլոր ուսանողներն են հավասարապես պատրաստ հեռավար ուսուցման իրականացմանը, ուստի պատրաստակամությունը չի կարող ամրագրվել նույն, խիստ հստակ մակարդակում: Հեռավար ուսուցում իրականացնելու պատրաստակամությունը կղիտարկվի որպես անձի կայուն դինամիկա, որը որոշում է նրա գործունեության բովանդակությունը, ուշադրությունը և բնույթը մասնագիտական ինքնորոշման և ինքնաիրացման գործընթացում: Հեռավար ուսուցման իրականացման պատրաստակամությունը չափելու խնդիրը կապված է դրա ձևավորման չափանիշների և մակարդակների նույնացման խնդրի հետ:

Հեռավար ուսուցման առաջարկվող համակարգում գնահատումը պետք է կատարվի նախանշված առաջադրանքների, ինքնուրույն աշխատանքների և դրանց կատարման փաստացի արդյունքների համապատասխանության հիման վրա: Առաջարկվող համակարգը պետք է ապահովի համապատասխան վերահսկողական ընթացակարգերի առկայություն և դրանց պարբերական ու կանոնավոր իրականացումը, այն դասախոսական անձնակազմին հնարավորություն է տալիս տեղեկանալու իր աշխատանքի կատարողականի մասին՝ հստակ տարանջատված թերություններով և ձեռքբերումներով:

Ուսուցման արդյունքների վերահսկումը կամ ստուգումը ուսուցման գործընթացի պարտադիր բաղադրիչն է: Այն տեղի է ունենում ուսումնական գործընթացի բոլոր փուլերում, բայց հատուկ նշանակություն է ձեռք բերում ծրագրի մի հատվածն ուսումնասիրելուց և վերապատրաստման փուլն ավարտելուց հետո: Ուսուցման արդյունքների ստուգման էությունը ուսանողների կողմից գիտելիքների տիրապետման մակարդակի բացահայտումն է, որը պետք է համապատասխանի տվյալ ծրագրի, առարկայի կրթական ստանդարտին [1]:

Հեռավար ուսուցման խնդիրներից մեկը քննությունների ժամանակ սովորողների կառավարման գործընթացն է:

Մեր կողմից նախագծված համակարգը նպատակաուղղված է սովորողների գիտելիքների, կարողությունների և հմտությունների ստուգման ու գնահատման առավել անաչառ, թափանցիկ ու արդյունավետ գործընթացի կազմակերպմանը:

Հեռավար ուսուցման առաջարկվող համակարգը պետք է հնարավորություն տա ստանալ անաչառ և ճշգրիտ տեղեկություն սովորողի գիտելիքի, մասնագիտական կարողունակությունների և պատրաստականության վերաբերյալ:

Առաջարկվող համակարգում անհրաժեշտ է հաշվի առնել գիտելիքի գնահատման այնպիսի պայմաններ, որը կնպաստի սովորողի և դասավանդողի աշխատանքի արդյունավետության բարձրացմանը:

Թեստը գործիք է, որը թույլ է տալիս բացահայտել յուրացման մակարդակը և որակը: Հարկ է նշել, որ ուսանողների գիտելիքների վերահսկման վիճակի ուսումնասիրությունը ստուգիչ հաշվիչների միջոցով որոշ խնդիրներ է հայտնաբերել թեստեր օգտագործելիս:

Հեռավար ուսուցման թեստերի համակարգ ստեղծելու ժամանակ ընտրելով բովանդակությունը՝ անհրաժեշտ է առաջնորդվել մի շարք սկզբունքներով [2].

- նյութի նշանակությունը,
- գիտական արժանահավատություն,
- թեստի բովանդակության համապատասխանությունը գիտության ներկա վիճակի մակարդակին,
- ներկայացուցչականություն (վերահսկողության համար բովանդակության տարրերի ամբողջականությունն ու բավարարությունը),
- աճող դժվարություն,
- հետևողականություն (փորձարկման առարկաների բովանդակության համապատասխանությունը գիտելիքների հետևողականության պահանջներին),
- բարդություն և հավասարակշռություն (հիմնական թեմաների համապարփակ ցուցադրում և հիմնական տեսական նյութի և գործնական գործունեության մեթոդների հավասարակշռված ցուցադրում),
- բովանդակության և ձևի փոխհարաբերություն:

Դրանք պետք է մշակվեն ուսումնառության յուրաքանչյուր մակարդակի համար:

Նախատեսվում է նաև հատուկ ուշադրություն դարձնել սովորողի ինքնուրույն ուսումնական աշխատանքի արդյունավետությունը [3]:

Հեռավար ուսուցման առաջարկվող համակարգում ուսումնական գործընթացն ավելի արդյունավետ դարձնելու նպատակով պետք է պարզեցնել մատուցվող նյութը՝ այն բաժանելով մասերի:

Համակարգում նախատեսվում է ստեղծել տվյալների բազա, որը հնարավորություն կտա իրականացնելու վերջնարդյունքների գնահատում:

Համակարգում կատարված է դասընթացներն ըստ գնահատման ձևի բաժանում երկու խմբի՝ քննական և ստուգարքային, ինչը հեռավար ուսուցման ավարտին ձեռք բերված ուսումնառության վերջնարդյունքների արժանահավատ չափումն է [4].:

Հեռավար ուսուցման առաջարկվող համակարգում քննություններին մասնակցելու համար անհրաժեշտ է գրանցվել առցանց՝ ընտրելով քննությունների օրերը: Առաջարկվող համակարգը հնարավորություն կտա զարգացնելու գնահատման և թեստավորման ժամանակակից՝

համակարգչային և առցանց ձևերը, որոնք ապահովում են գնահատման օբյեկտիվությունը, արդարությունը, համադրելիությունը և ավտոմատացումը՝ միաժամանակ կրճատելով դասախոսի սուբյեկտիվ գործոնի ազդեցությունը և գնահատման վրա կատարվող ծախսերը [5]:

Այսպիսով, ՏՀՏ գործիքների միջոցով գիտելիքների ստուգման և գնահատման առցանց հնարավորությունները, կախված դասի տեսակից և ընտրված մեթոդից, ուսումնական գործընթացը կդարձնեն առավել հետաքրքիր և արդյունավետ, ինչն էլ կխթանի դասավանդողների և սովորողների համատեղ գործունեության արդյունավետ զարգացմանը:

Ստորև ներկայացնում ենք թեստերի անցման կարգավորման ֆունկցիոնալությունից ծրագրային կոդի մի հատված:

```
[Authorize(Roles = "Professor")]
```

```
public class TestConfigurationsController : Controller
{
    private readonly ApplicationDbContext _context;
    private readonly UserManager<ApplicationUser>
_userManager;

    public
TestConfigurationsController(ApplicationDbContext context,
UserManager<ApplicationUser> userManager)
    {
        _context = context;
        _userManager = userManager;
    }

    // GET: TestConfigurations/Index
    public async Task<IActionResult> Index()
    {
        var testConfigurations = await
_context.TestConfigurations
.Include(tc => tc.Section)
```

```
        .Where(tc => tc.CreatedById ==
_userManager.GetUserId(User))
        .OrderBy(tc => tc.Created)
        .ToListAsync();
    return View(testConfigurations);
}

// GET: TestConfigurations/Details/5
public async Task<IActionResult> Details(string id)
{
    if (id == null)
    {
        return NotFound();
    }

    var testConfig = await _context.TestConfigurations
        .Include(tc => tc.Section)
        .Include(tc => tc.TCQuestions)
        .ThenInclude(tcq => tcq.Question)
        .FirstOrDefaultAsync(t => t.Id == id);

    if (testConfig == null)
    {
        return NotFound();
    }

    return View(testConfig);
}
```

```
// GET: TestConfigurations/Results/5
public async Task<IActionResult> Results(string id)
{
    if (id == null)
    {
        return NotFound();
    }

    var test = await _context.Tests
        .Include(t => t.Student)
        .ThenInclude(s => s.Base)
        .Include(t => t.TestQuestions)
        .ThenInclude(tq => tq.Question)
        .ThenInclude(q => q.Answers)
        .FirstOrDefaultAsync(t => t.Id == id);

    if (test == null)
    {
        return NotFound();
    }

    return View(test);
}

// GET: TestConfigurations/Create
[HttpGet]
public async Task<IActionResult> Create()
```

```

    {
        ViewData["Sections"] = await GetSections();
        return View(new TestConfigurationViewModel());
    }

    // POST: TestConfigurations/Create
    [HttpPost]
    public async Task<IActionResult>
Create([Bind("Name,SectionId,Duration,RandomQuestions," +
"QuestionsCount,QuestionsFrom,QuestionsTo,ActiveFrom," +
"ActiveTo,ActiveFromTime,ActiveToTime,Questions")]
TestConfigurationViewModel viewModel)
{
    if (ModelState.IsValid)
    {
        var testConfiguration = new TestConfiguration
        {
            Name = viewModel.Name,
            SectionId = viewModel.SectionId,
            Duration = viewModel.Duration,
            CreatedById = _userManager.GetUserId(User),
            Created = DateTime.Now,
            TCQuestions = new
List<TestConfigurationQuestion>()

        };

        if
(!string.IsNullOrEmpty(viewModel.ActiveFromTime) &&

```

```
!string.IsNullOrEmpty(viewModel.ActiveToTime))
    {
        var hhFrom =
Convert.ToInt32(viewModel.ActiveFromTime.Substring(0, 2));
        var mmFrom =
Convert.ToInt32(viewModel.ActiveFromTime.Substring(3, 2));
        var hhTo =
Convert.ToInt32(viewModel.ActiveToTime.Substring(0, 2));
        var mmTo =
Convert.ToInt32(viewModel.ActiveToTime.Substring(3, 2));
        testConfiguration.ActiveFrom =
viewModel.ActiveFrom.Date.AddHours(hhFrom).AddMinutes(mmFrom);
        testConfiguration.ActiveTo =
viewModel.ActiveTo.Date.AddHours(hhTo).AddMinutes(mmTo);
    }
    else
    {
        testConfiguration.ActiveFrom =
viewModel.ActiveFrom.Date;
        testConfiguration.ActiveTo =
viewModel.ActiveFrom.Date;
    }

    testConfiguration.RandomQuestions =
viewModel.RandomQuestions;
    if (viewModel.RandomQuestions)
    {
        testConfiguration.OnlySelected =
viewModel.OnlySelected;
        testConfiguration.QuestionsCount =
viewModel.QuestionsCount;
```

```

        testConfiguration.QuestionsFrom =
viewModel.QuestionsFrom;

        testConfiguration.QuestionsTo =
viewModel.QuestionsTo;

        if (viewModel.Questions != null)
        {
            foreach (var tcq_vm in
viewModel.Questions)
            {
                var tcq = new
TestConfigurationQuestion
            {
                QuestionId = tcq_vm.Id,
                Status = tcq_vm.Status
            };

testConfiguration.TCQuestions.Add(tcq);
        }
    }
}
else
{
    if (viewModel.Questions != null)
    {
        foreach (var tcq_vm in viewModel.Questions.Where(q => q.Status
== TCQStatus.Include))
        {
            var tcq = new
TestConfigurationQuestion
            {

```

```
                QuestionId = tcq_vm.Id,
                Status = tcq_vm.Status
            };

testConfiguration.TCQuestions.Add(tcq);
    }

    testConfiguration.QuestionsCount =
(byte)testConfiguration.TCQuestions.Count;
    }
    else
    {
        ModelState.AddModelError("Questions", "You doesn't
select any question.");

        return View(viewModel);
    }
}

_context.TestConfigurations.Add(testConfiguration);
    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}

return View(viewModel);
}

// GET: TestConfigurations/Questions
[HttpPost]
public async Task<IActionResult> Questions(string
sectionId,
```

```

        [Bind("Type,AnswersCount,Page,Exist")]
        TCQRequestViewModel viewModel)
    {
        if (string.IsNullOrEmpty(sectionId))
            return NotFound();

        if (viewModel.Page < 0)
            viewModel.Page = 0;

        var query = _context.Questions.Where(q =>
q.SectionId == sectionId);
        if (viewModel.Type != QuestionType.All)
        {
            query = query.Where(q => q.Type ==
viewModel.Type);
        }
        if (viewModel.AnswersCount > 0)
        {
            query = query.Where(q => q.Answers.Count ==
viewModel.AnswersCount);
        }
        var count = await query.CountAsync();
        if (count / 20 < viewModel.Page)
            viewModel.Page = count / 20;
        var result = await query.Skip(viewModel.Page *
20).Take(20).ToListAsync();

        viewModel.Total = count;
        viewModel.Questions = result;
    }

```

```
        return PartialView(viewModel);
    }

    private async Task<SelectList> GetSections()
    {
        var sections = await (from p in _context.Employes
                                join c in _context.Chair on
                                p.Chair.Id equals c.Id
                                join s in _context.Section on
                                c.Id equals s.ChairId
                                select s).ToListAsync();
        return new SelectList(sections, "Id", "Name");
    }

    // GET: Questions/Delete/5
    public async Task<IActionResult> Delete(string id)
    {
        if (id == null)
        {
            return NotFound();
        }

        var testConfiguration = await
            _context.TestConfigurations
                .Include(tc => tc.Section)
                .FirstOrDefaultAsync(tc => tc.Id == id);
        if (testConfiguration == null)
        {

```

```
        return NotFound();
    }

    return View(testConfiguration);
}

// POST: Questions/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public async Task<IActionResult> DeleteConfirmed(string
id)
{
    var testConfiguration = await
_context.TestConfigurations.FindAsync(id);

_context.TestConfigurations.Remove(testConfiguration);
    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}
}
}
```

Հոդվածը տպագրվում է ԱՀ կառավարության կողմից ֆինանսավորվող Արցախի գիտական կենտրոնի հետ համատեղ իրակացվող **«18 AK-004 "Արցախում հեռավար ուսուցման WEB ծրագրային մոդելի մշակում"»** գիտական թեմայի շրջանակներում:

Գրականություն

1. Раззаков, Ш. И., Контроль знаний в системе дистанционного обучения. Молодой ученый. — 2014. № 7 (66). — С. 70-73
2. Канаев В. И. Дистанционное обучение: технологические аспекты. М.: Современный гуманитарный университет: 2004. — 192 с.
3. Марина Ефимовна Вайндорф-Сысоева., Методика дистанционного обучения. Учебное пособие для вузов- ЛитРес: 2017-195 с.
4. Теория и практика дистанционного обучения: Учебное пособие для студентов высших педагогических заведений / Е.С. Полат, М.Ю. Бухаркина, М.В. Моисеева; Под ред. Е.С. Полат. - М.: Издательский центр «Академия»: 2004. - 416 с.
5. Вайндорф - Сысоева М.Е. Методика дистанционного обучения. Учебное пособие для вузов- издательство Юрайт., 2020г.-194 с.

Հոդվածը տպագրության է երաշխավորել խմբագրական խորհրդի անդամ, ֆ.մ.գ.թ. Գ. Ն.Սահակյանը: