УДК 794.242(07)
DOI https://doi.org/10.31392/UDU-nc.series5.2025.105.07

Губарєв І. В., Іскандарова-Мала А. О., Музичка К. О.

РОЗРОБКА ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ГРИ В ШАШКИ-64 З МЕТОЮ УДОСКОНАЛЕННЯ ТАКТИЧНИХ ТА ТЕХНІЧНИХ НАВИЧОК СПОРТСМЕНІВ-ШАШКІСТІВ

Зі зростанням майстерності спортсмена, індивідуалізація в навчально-тренувальному процесі, особливо в інтелектуальних видах спорту, набуває великої значущості.

Для реалізації цього створено комп’ютерну навчально-тренувальну ігрову програму “Draughts” для двох гравців (тренер-викладач з шашок – спортсмен-вихованець; спортсмен-вихованець – спаринг-партнер («рівний по підготовці суперник в тренуванні»)) за допомогою крос-платформного інструментарію розробки програмного забезпечення Qt мовою програмування С++.

Застосунок розроблено за допомогою об’єктно-орієнтованого програмування (ООП). Надано повний опис роботи програми, її алгоритм, описано як користуватися програмою. Розроблено інтерфейс зручний та зрозумілий усім користувачам: тренеру з шашок, спортсмену, спаринг-партнеру.

Об’єктно-орієнтована програма складається з об’єктів – окремих фрагментів коду, що обробляє дані, які взаємодіють один з одним через певні інтерфейси, забезпечуючи дотримування відомих методичних правил наступності завдань: – від простого до складного, від відомого до невідомого. Після запуску на виконання програми користувач бачить дошку для шашок-64 з уже розставленими на ній шашками білого та чорного кольорів. Щоб зробити хід, гравцеві необхідно натиснути на шашку його кольору, а потім на якесь із місць, що запропонує комп’ютер.

Усі можливі варіанти ходу для вибраної шашки пропадуть. Якщо можливе взяття шашки суперника, то бій за правилами гри є обов’язковим, і програма не дозволить зробити звичайний хід.

Ідеї ООП знайшли застосування у підвищенні ефективності навчання шашковій грі за рахунок більш конкретного і правильного уявлення про тактичні дії проти можливого суперника, або підготовки і проведення стандартних шашкових комбінацій на основі зорового і чуттєвого сприйняття.

Ця програма дозволяє під час гри зберігати хід подій спільної партії гравців, що потрібно тренеру, щоб зробити оцінку тактичних та технічних дій своїх підопічних. Завдяки цьому вихованці зможуть в аналізах зрозуміти, чому їхній суперник вибрав те чи інше продовження у шашковій грі.

Ключові слова: *шашки, шашкова дошка, класи, об’єкти, тренер-викладач, гравець, тактичні дії, об’єктно-орієнтоване програмування.*

У системі багатолітньої підготовки юних шашкістів процес тренування та змагань органічно зв’язані та розглядаються як категорії єдиного педагогічного процесу. Як у навчанні, так і у турнірах, вирішуються питання та задачі удосконалення тактичних та технічних навичок у грі. Реалізується навчання шляхом раціонального підбору технічних прийомів (позицій) з урахуванням можливостей і теоретичної підготовки юних шашкістів [1]. Зі зростанням майстерності шашкіста (його розряду) індивідуалізація в навчальному процесі набуває великої значущості (тренувальні партії з тренером-наставником педагогом).

Основою даної роботи стало створення комп’ютерної навчально-тренувальної ігрової програми “Draughts” для двох гравців (тренер-викладач з шашок – спортсмен-вихованець; спортсмен –вихованець – спаринг-партнер («рівний по підготовці суперник в тренуванні») за допомогою крос-платформного інструментарію розробки програмного забезпечення Qt мовою програмування С++ [2; 3]. Застосунок створено за допомогою об’єктно-орієнтованого програмування (однієї з парадигм програмування, яка розглядає програму як множину «об’єктів», що взаємодіють між собою), основна мета якого, як і більшості інших підходів до програмування – підвищення ефективності розробки програм.

Ідеї ООП виявилися плідними й знайшли застосування у підвищенні ефективності навчанню шашкової гри за рахунок більш конкретного і правильного уявлення про тактичні дії проти можливого суперника, або підготовки і проведення стандартних шашкових комбінацій на основі зорового і чуттєвого сприйняття.

Постановка задачі: розробити алгоритм шашкової гри. Використовуючи ООП, написати програму мовою С++ у вигляді сукупності взаємодіючих об’єктів, кожен з яких є екземпляром певного класу, а класи є членами певної ієрархії наслідування, яка реалізує принцип гри в шашки-64 та допомагає прискорити оволодіння шашковими прийомами.

Метою статті є створення застосунку, що представляє собою програмну реалізацію відомої логічної гри “Draughts”; засвоювання методики експериментальних досліджень, оволодіння навичками співставлення результатів своїх досліджень із літературними даними, аналізу, узагальнення і оформлення одержаних результатів.

Методи досліджень – аналіз предметної області гри “Draughts” та вимог до її програмного забезпечення (властивості об’єктно-орієнтованого програмування, математичний інструментарій логістики, алгоритми та структури даних).

Кожний тренер-викладач, розпочинаючи навчання юних любителів шашкової гри, завжди вирішує проблемне завдання під питанням: «З чого почати? Як і якими засобами пояснити початківцям спортсменам закони та правила гри, розкрити її красу та глибину змісту?»

Ефективність навчально-тренувального процесу завжди базується на дотриманні деяких методичних принципів: наочність, систематичність, доступність, індивідуальність, прогресування. Розрізняють безпосередню та опосередковану наочність. До першої відноситься безпосередній показ дії, що вивчається або вдосконалюється (наприклад у легкій партії, під час проведення комбінації), до другої – позиції на діаграмах, демонстраційних дошках та ін.

Для ефективного вирішення питань наочності, з метою підготовки спортсменів-розрядників, було створено програму “Draughts” [6], яка є симулятором логічної гри шашки-64, у ході якої користувачі, що грають білим або чорним кольором, мають вибрати необхідну шашку, а потім напрямок її ходу. Далі система вказує всі можливі ходи, перевіряє правильність ходу і робить його. Інтерфейс розробленої програми представлений на рисунку 1.

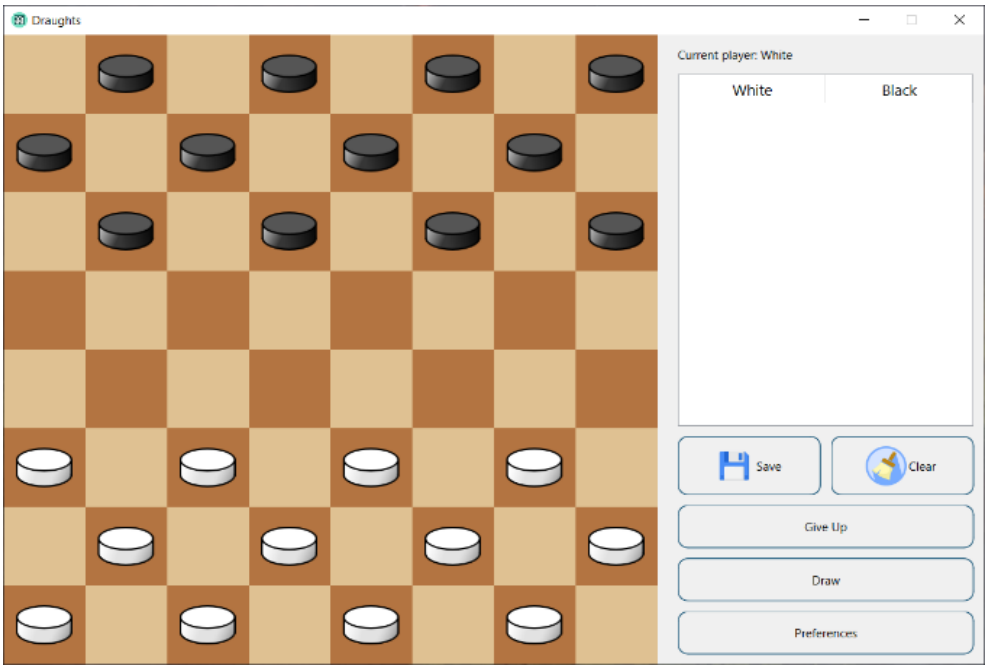


Рис. 1. Вікно програми “Draughts”

У шашковій грі є два типи фігур – прості і дамки. У початковому положенні усі шашки прості. Розстановка початкової позиції показана на рисунку 1. Коли проста шашка досягає останню горизонталь (у напрямку руху), вона перетворюється в інший тип фігури – дамку (рис. 2).

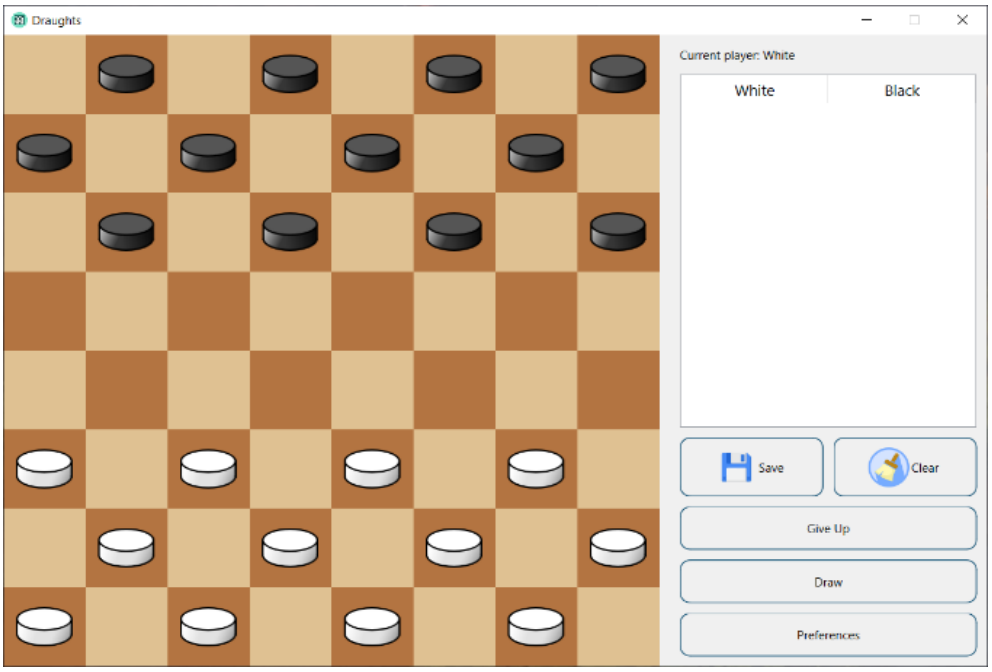


Рис. 2. Дамки

Дамка – це шашка, що доведена до останнього горизонтального ряду клітин противника з отриманням права пересуватися на будь-яке число клітин в будь-якому напрямку. Тому, якщо шашка суперника досягає останньої горизонталі (у напрямку руху), її зовнішній вигляд змінюється (шашка стає подвійною), і користувач розуміє, що це – дамка (рис. 2).

Обидва гравці роблять свої ходи по черзі: за один раз один хід. Переміщення своєї фігури на вільне ігрове поле, без взяття фігури суперника або з ним, визначається як хід у партії. Перший хід у початковому положенні робить той, кому за жеребом, або за будь-якої попередньої домовленості, випало грати білими. Прості білі шашки ходять вгору дошки, якщо «низом» приймати їх початкове положення, на одну клітку по діагоналі. Дамки ходять по діагоналі на будь-яке поле, як вперед, так і назад, причому перескакувати вони можуть тільки шашки суперника з дотриманням правила — за кожною шашкою суперника повинно бути вільне поле.

Реалізація програми відбувається за допомогою мови програмування C++ з використанням базового функціоналу об’єктно-орієнтованого програмування, де основними є поняття об’єктів і класів [4; 5].

При натисканні на кнопку “Preferences” (рис. 3) отримаємо вікно, у якому побачимо три кнопки, за допомогою яких можна змінити колір темної та світлої клітинок, а також розмір вікна, вибравши його перед цим за допомогою слайдера.

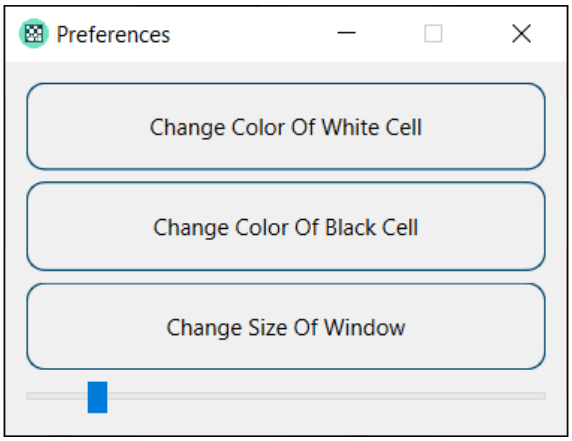


Рис. 3. Вікно налаштувань

Після запуску на виконання програми користувач бачить дошку для шашок-64 з уже розставленими на ній шашками білого та чорного кольорів. Щоб зробити хід, гравцеві необхідно натиснути на шашку його кольору, а потім на якесь із місць, що запропонує комп’ютер (рис. 4).

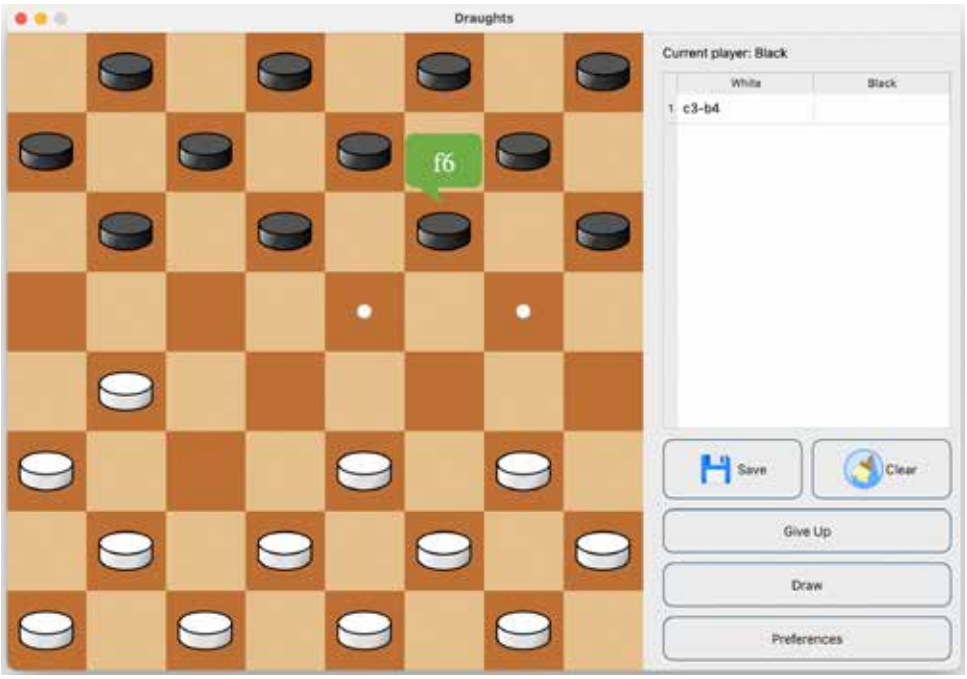


Рис. 4. Пропонування комп’ютером ходів для гравця, що натиснув на чорну шашку f6

Взагалі кожна клітинка дошки є кнопкою, при натисканні на яку, викликається функція реалізації ходу гравця. Кожна кнопка-клітинка може містити зображення у вигляді шашки. Кожне ігрове поле дошки має двозначний індекс. Горизонтальні ряди шашкової дошки позначаються цифрами від 1 до 8, а вертикальні – латинськими буквами від a до h. Хід позначається двома полями. Наприклад, якщо шашка пересувається з поля c3 на поле b4, хід позначається наступним чином: c3–b4 (рис. 4).

Хід із взяттям записується у вигляді позначень полів, розділених двокрапкою, за якими проходить шашка, що забирає, наприклад (рис. 2): запис 2.b4:d6 означає хід білих з поля b4 на поле d6 зі взяттям чорної шашки суперника, що стоїть на полі c5.

Для зміни позиції шашки, якою хоче здійснити хід гравець, необхідно натиснути на будь-яке місце дошки, що пропонує програма. Усі можливі варіанти ходу для вибраної шашки пропадуть. Якщо можливе взяття шашки суперника, то бій за правилами гри є обов'язковим, і програма не дозволить зробити звичайний хід.

Щоб розпочати нову гру необхідно натиснути на кнопку "Draw" або "Give Up". На дошці автоматично з'явиться початкове положення шашок.

Взяття шашки суперника аналогічне звичайному ходу. Спочатку необхідно обрати шашку, якою хочемо зробити удар, а потім натиснути на запропоноване програмою місце на дошці. Якщо можливе продовження удару, тобто є можливість взяття декількох шашок суперника, то необхідно знову натиснути на шашку, якою робимо взяття, щоб побачити можливе продовження удару.

Закінчити гру можна натиснувши на будь-яке місце на дошці за умови, що в одного з гравців усі шашки зняті з дошки або є запертими (перемога). За відсутності такої умови, щоб закінчити партію нічиєю, треба натиснути на кнопку "Draw".

Для зміни параметрів програми необхідно натиснути кнопку "Preferences", в отриманому вікні налаштовується колір клітинок на дошці та розмір вікна додатку.

Дана програма дозволяє під час гри гравців зберігати хід подій їх спільної партії, що потрібно тренеру, щоб зробити оцінку тактичних та технічних дій своїх підопічних. Завдяки цьому вихованці зможуть в аналізах зрозуміти, чому їхній суперник вибрав те чи інше продовження у шашковій грі. Аналіз зіграних турнірних партій із практичними коментарями тренера, зміцнюють віру спортсмена у свої сили, доставляють їм радість творчості та естетичне задоволення.

До аналізу тренер має готуватися особливо ретельно, вивчаючи запис партії. Під час розбору позицій необхідно вказати ступінь виконання стратегічного, тактичного та дебютного плану. До аналізу партій та позицій залучаються усі без винятку члени команди та учасники змагань. Вміння тренера-викладача керувати змагальною діяльністю під час шашкових турнірів своїх підопічних складається: з уміння спостерігати, робити аналіз різних дебютних схем противника, помилок вихованців під час партій, з прийняття рішень у вигляді конкретних дебютних установок, домашніх аналізів зіграних партій.

Об'єктно-орієнтована програма складається з об'єктів – окремих фрагментів коду, що обробляє дані, які взаємодіють один з одним через певні інтерфейси, забезпечуючи дотримування відомих методичних правил: наступності завдань (позицій) та занять – від простого до складного, від відомого до невідомого.

Дана розробка допомагає у забезпеченні ясності з розвитку мислення та розуміння шашкових проблем, усіх тонкощів тактики та стратегії у вивченні гри, що постають перед спортсменами-вихованцями.

З її допомогою тренером вирішуються також такі завдання:

- виховання вольових якостей у шашкістів під час гри;
- розвиток у початківців процесів сприйняття «почуття небезпеки» та «почуття часу»;
- розвиток уваги, її стійкості, зосередженості під час гри;
- формулювання ігрової установки на кожну партію кожному вихованцю окремо.

Висновки. Необхідно помітити, що ООП не є панацеєю від всіх програмістських лих, але його цінність як передової технології програмування безсумнівна. Вивчення ідей і методів ООП може істотно спростити розробку й налагодження складних програм.

Під час виконання роботи були відпрацьовані основні етапи створення програми: постановка задачі, розробка алгоритму, складання програми, її встановлення на ПК, налагодження та тестування її функцій, перевірка відповідності правилам гри у шашки.

Був створений повний опис роботи програми, її алгоритм, описано як користуватися програмою. Було проаналізовано дану область та знайдено потрібну інформацію по даній тематиці. Розроблено інтерфейс зручний та зрозумілий усім користувачам: тренеру-викладачу з шашок, спортсмену-вихованцю, спаринг-партнеру.

Програма допомагає не лише навчити грати у шашки та підняти кваліфікацію чинному спортсмену, а й створює можливість тренеру навчити думати своїх підопічних у певних типах позицій та стадіях партій, об'єктивно оцінювати ступінь виконання поставленого завдання кожним вихованцем.

Використана література:

1. Губарєв І. В., Савінський Д. О. Практичний курс занять з шашкового спорту : навч. посіб. Кам'янське : ДДТУ, 2018. 159 с.
2. Bjarne Stroustrup. The C++ Programming Language: Addison-Wesley Professional. 4th edition, 2013. 1368 p.
3. Herbert Schildt. C++: The Complete Reference: McGraw Hill. 4th Edition, 2002. 1056 p.

4. Шпак З. Я. Програмування мовою С : навч. посіб. Вид. 2-ге, допов. Львів : *Видавництво Львівської політехніки*, 2011. 436 с.
5. Яшина О. В., Жульковський О. О. Методичні вказівки до виконання лабораторних робіт з дисципліни «Мови об’єктно-орієнтованого програмування» для студентів спеціальності 6.080403 «Програмне забезпечення автоматизованих систем». Дніпродзержинськ : ДДТУ, 2004. 63 с.
6. Губарєв І. В., Іскандарова-Мала А. О., Музичка К. О. Розробка комп’ютерної навчально-тренувальної ігрової програми “Draughts” для підвищення ефективності підготовки спортсменів. Перспективи стабільного економічного розвитку та соціальних комунікацій в умовах сучасних викликів: вітчизняні реалії та світовий досвід : матеріали І Всеукр. наук.-практ. конф. з міжнародною участю (присвяченої 30-річчю економічного факультету ДДТУ) м. Кам’янське, 25–26 квітня 2024 р. / ред. кол.: Н. А. Караван [та ін.]. Кам’янське : ДДТУ, 2024. С. 571–575.

References:

1. Hubariev I. V., Savinskyi D. O. (2018). Praktychnyi kurs zaniat z shashkovoho sportu [Practical course of classes in checkers] : navch. posib. Kam’ianske : DDTU. 159 s. [in Ukrainian].
2. Bjarne Stroustrup. (2013). The C++ Programming Language, Addison-Wesley Professional; 4th edition. 1368 p.
3. Herbert Schildt. (2002). C++: The Complete Reference, McGraw Hill, 4th Edition. 1056 p.
4. Shpak Z. Ya. (2011). Prohramuvannya movoyu S [Programming in the C language] : navch. posib. Vyd. 2-he, dopov. L'viv : *Vydavnytstvo L'vivs'koyi politekhniky*. 436 s. [in Ukrainian].
5. Yashyna O. V., Zhul'kovs'kyy O. O. (2004). Metodychni vkazivky do vykonannya laboratornykh robit z dystsypliny “Movy ob’yektno-oriyentovanoho prohramuvannya” dlya studentiv spetsial'nosti 6.080403 “Prohramne zabezpechennya avtomatyzovanykh system” [Methodological instructions for performing laboratory work in the discipline “Object-oriented programming languages” for students of the specialty 6.080403 “Software of automated systems”]. Dniprodzerzhyns'k : DDTU. 63 s. [in Ukrainian].
6. Hubaryev I. V., Iskandarova-Mala A. O., Muzychka K. O. (2024). Rozrobka komp’yuternoyi navchal'no-trenuval'noyi ihrovoyi prohramy “Draughts” dlya pidvyshchennya efektyvnosti pidhotovky sportsmeniv [Development of the computer educational and training game program “Draughts” to increase the efficiency of training of athletes]. Perspektyvy stabil'noho ekonomichnoho rozvytku ta sotsial'nykh komunikatsiy v umovakh suchasnykh vyklykiv: vitchyznyani realiyi ta svitovyy dosvid : materialy I Vseukr. nauk.-prakt. konf. z mizhnarodnoyu uchastyu (prysvyachenoyi 30-richchyu ekonomichnoho fakul'tetu DDTU). m. Kam’yans'ke, 25–26 kvitnya 2024 r./red. kol.: N. A. Karavan [ta in.]. Kam’yans'ke : DDTU. S. 571–575 [in Ukrainian].

I. Hubariev, A. Iskandarova-Mala, K. Muzychka. Development of game information and software in checkers-64 in order to improve the tactical and technical skills of athletes – checkers.

With the growth of the athlete's skill, individualization in the educational and training process, especially in intellectual sports, becomes of great importance.

To implement this, a computer educational and training game program “Draughts” was created for two players (checkers trainer/teacher – athlete-student; athlete-student –sparring partner (“equally trained opponent in training”) using cross-platform development tools Qt software in the C++ programming language.

The application was developed using object-oriented programming (OOP). A full description of the program's operation, its algorithm, and how to use the program are provided. The interface has been developed to be convenient and understandable for all users: to the checkers coach, athlete, sparring partner.

An object-oriented program consists of objects -separate fragments of code that process data, which interact with each other through certain interfaces, ensuring compliance with well-known methodological rules of task sequence: – from simple to complex, from known to unknown.

After starting the program, the user sees a 64-checker board with white and black checkers already placed on it. To make a move, the player must click on a checker of his color, and then on one of the places suggested by the computer.

All possible move options for the selected checker will disappear. If it is possible to capture the opponent's checker, then a fight according to the rules of the game is mandatory, and the program will not allow you to make a normal move.

The ideas of OOP have found application in increasing the effectiveness of learning the game of checkers due to a more specific and correct idea of tactical actions against a possible opponent, or preparation and implementation of standard checkers combinations based on visual and sensory perception.

This program allows you to save the course of events of a joint game of players during the game, which is necessary for the coach to evaluate the tactical and technical actions of his wards. Thanks to this, the students will be able to understand in the analysis why their opponent chose a particular continuation in the game of checkers.

Key words: *checkers, board, classes, objects, trainer-teacher, player, tactical actions, object-oriented programming*