

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

**«Московский государственный технический университет радиотехники, электроники и автоматики»**

**МГТУ МИРЭА**

***Институт информационных технологий (ИТ)***

«наименование факультета»

Кафедра базовая автоматизированных систем организационного управления (**АСОУ**) **№239** МГТУ МИРЭА при ФГУП НИИ «Восход»

«наименование кафедры»

**КУРСОВАЯ РАБОТА**

**по дисциплине**

\_\_\_\_\_Программирование 1\_\_\_\_

<Наименование дисциплины>

**Тема курсовой работы:** «Игры пошаговые: Морской бой»

Предметная область: «Игры пошаговые»

<Название темы курсовой работы/проекта>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Студент группы ИВБ-3-14 |  | |  | Аллямов И.З. | | |
| <код группы> | <подпись студента> | |  | <ФИО студента> | | |
| Руководитель курсовой работы |  | |  | Ассистент | | |
|  |  | |  | <должность, звание, уч.степень> | | |
|  |  | |  | Хлебников А.А. | | |
|  | <подпись руководителя> | |  | < ФИО руководителя > | | |
| Рецензент *(при наличии)* |  | |  |  | | |
|  |  | |  | <должность, звание, уч.степень> | | |
|  |  | |  |  | | |
|  | <подпись рецензента> | |  | < ФИО рецензента> | | |
| *Работа представлена к защите* | | | «\_\_\_»\_\_\_\_\_\_\_\_201\_ г. | | |  |  | |
|  | | |  | | |  | <подпись студента> | |
| *«Допущен к защите»* | | | «\_\_\_»\_\_\_\_\_\_\_\_201\_ г. | | |  |  | |
|  | | |  | | |  | <подпись руководителя> | |



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

**«Московский государственный технический университет радиотехники, электроники и автоматики»**

**МГТУ МИРЭА**

***Институт информационных технологий (ИТ)***

«наименование факультета»

Кафедра базовая автоматизированных систем организационного управления (**АСОУ**) **№239** МГТУМИРЭА при ФГУП НИИ «Восход»

«наименование кафедры»

|  |  |
| --- | --- |
|  | **УТВЕРЖДАЮ**  Заведующий  кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.В. Волков  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 201\_\_г. |

**ЗАДАНИЕ**

**на выполнение курсовой работы**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **по дисциплине** | | «Программирование 1» | | | |
|  | | «Название дисциплины» | | | |
| Студент | Аллямов Ильдар Зиннятуллаевич | | Группа | ИВБ-3-14 |
|  |  | |  |  |

**1.Тема курсовой работы**

«Игры пошаговые: Морской бой»

**2. Исходные данные**

* Текстовый интерфейс игры:
  + - Игровой процесс
    - Текстовое отображение результата
* Программа должна обеспечивать следующие операции:
  + - Беспроблемная работа программы
    - Простота использования программы
    - Адекватное отображение результата использования программы

**2. Перечень вопросов, подлежащих разработке, и обязательного графического материала:**

* Разработка текстового интерфейса и базовых функций игры «Морской бой».
* Программная реализация игры «Морской бой».

**4. Срок представления к защите курсового проекта (работы):** до «\_\_\_» \_\_\_\_\_\_\_\_\_\_201\_ г.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Задание на курсовой проект (работу) выдал | «\_\_\_»\_\_\_\_\_\_\_201\_ г. |  |  |  | Хлебников А.А. |
|  |  |  | <подпись руководителя проекта> |  | <Ф.И.О. руководителя проекта> |
| Задание на курсовой проект (работу) получил | «\_\_\_»\_\_\_\_\_\_\_201\_ г. |  |  |  | Аллямов И.З. |
|  |  |  | <подпись студента-исполнителя проекта> |  | <Ф.И.О. студента-исполнителя проекта> |

**Мониторинг процесса выполнения курсового проекта (работы)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| №  э  т  апа | Наименование этапа курсового проекта, работы | Этап курсового проекта, работы выполнил и представил результаты руководителю проекта (работы),  *дата и подпись исполнителя* | Работу по этапу курсового проекта (работы) принял на рассмотрение,  *дата и подпись руководителя* | Рекомендации и замечания по этапу курсового проекта (работы) выдал исполнителю,  *дата и подпись руководителя* | Оценка выполнения этапа курсового проекта, (работы)*(в соответствии с балльно-рейтинговой системой)* | Комментарии руководителя курсового проекта (работы) |
| 1 | Разработка задания на КР |  |  |  |  |  |
| 2 | Анализ задачи |  |  |  |  |  |
| 3 | Разработка алгоритма задачи |  |  |  |  |  |
| 4 | Разработка структуры проекта |  |  |  |  |  |
| 5 | Разработка интерфейса пользователя |  |  |  |  |  |
| 6 | Разработка схемы программных взаимодействия единиц, группируя по модулям |  |  |  |  |  |
| 7 | Разработка приложения |  |  |  |  |  |
| 8 | Разработка пояснительной записки |  |  |  |  |  |
| 9 | Защита курсовой работы |  |  |  |  |  |

**Реферат**

**Объект исследования** – программирование на языке С++.

**Предмет исследования** – Игры пошаговые.

**Цель работы**: закрепление, углубление и использование практических знаний по программированию на языке С++, возможность разработки и реализации программы, работа с функциями и стандартными библиотеками C++. В процессе данной работы проводилась тщательная разработка алгоритмов, тестирование и отладка приложения, его элементов и модулей.

**Область возможного практического применения:** домашнее использование; пример для более детальной разработки схожих программ и приложений.

**ЛИСТ ЗАМЕЧАНИЙ**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Содержание

[Введение 6](#_Toc421060285)

[**История языка C++** 6](#_Toc421060286)

[**Игра «Морской бой»** 8](#_Toc421060287)

[Основная часть. 12](#_Toc421060288)

[**1.1 Средства для выполнения задания.** 12](#_Toc421060289)

[**1.2 Методы для выполнения задания.** 12](#_Toc421060290)

[**2.1 Интерфейс командной строки.** 14](#_Toc421060291)

[Заключение 17](#_Toc421060292)

[Список литературы 18](#_Toc421060293)

[Приложение 19](#_Toc421060294)

# 

# **Введение**

## **История языка C++**

C++ был изобретен Бьярном Страуструпом в 1979 г. в Bell Laboratories в Муррей Хилл, штат Нью-Джерси. Сначала Страуструп назвал новый язык “С с классами”. Однако в 1983 г. язык получил название C++. Страуструп построил C++ на базе С, сохранив все возможности С, его характерные особенности и достоинства. Он перенял также и базовую стратегию С, согласно которой программист, а не язык определяет структуру программы. Важно понимать, что Страуструп не создал совершенно новый язык программирования. Напротив, он расширил уже имеющийся весьма успешный язык. Большая часть средств, добавленных Страуструпом к С, предназначена для поддержки объектно-ориентированного программирования. В сущности, C++ является объектно-ориентированной модификацией С. Построив новый язык на базе С, Страуструп обеспечил плавный переход к внедрению ООП. Вместо того, чтобы изучать совершенно новый язык, программисту, работающему на С, было достаточно освоить лишь несколько новых средств, после чего он мог пожинать все плоды методологии объектно-ориентированного программирования. Создавая C++, Страуструп понимал, что добавляя к языку С поддержку объектно-ориентированного программирования, необходимо было в то же время сохранить дух языка С, включая его эффективность, гибкость и стратегию программирования. К счастью, эта цель была достигнута. C++ предоставляет программисту все возможности С, добавляя к ним могущество объектов.

Хотя первоначально C++ создавался в качестве средства разработки очень больших программ, его использование отнюдь не ограничивается этой сферой. В действительности объектно-ориентированные средства C++ могут эффективно применяться для решения практически любых задач программирования. Легко найти примеры разработки на C++ таких программных проектов, как текстовые или графические редакторы, базы данных, персональные файловые системы, сетевые утилиты и коммуникационные программы. В силу того, что C++ воспринял эффективность С, его часто используют для разработки высокопроизводительного системного программного обеспечения. Он также широко используется и для программирования в системе Windows.

С момента своего появления C++ претерпел три серьезных модификации, каждая из которых расширяла и изменяла язык. Первая модификация была предложена в 1985 г., вторая – в 1990 г. Третья модификация появилась в процессе стандартизации C++. Несколько лет назад началась разработка стандарта для C++. С этой целью организациями ANSI (Американский национальный институт стандартов) и ISO (Международная организация по стандартизации) был создан объединенный комитет по стандартизации. Первый вариант предлагаемого стандарта был выпущен 25 января 1994 г. В этот вариант комитет ANSI/ISO по C++ (членом которого я являлся) включил средства, определенные Страуструпом, с добавлением некоторых новых. В целом первый вариант стандарта отражал состояние C++ на тот момент времени. Вскоре после завершения работы над первым вариантом стандарта C++ произошло событие, которое заставило серьезно расширить стандарт: создание Александром Степановым стандартной библиотеки шаблонов (Standard Template Library, STL). STL представляет собой набор обобщенных процедур, с помощью которых можно обрабатывать данные. Эта библиотека сочетает в себе могущество и элегантность. Но она также весьма велика. Выпустив первый вариант стандарта C++, комитет проголосовал за включение STL в спецификацию C++. Добавление STL расширило возможности C++ далеко за пределы его первоначального определения. С другой стороны, включение STL, хотя и вполне оправданное, замедлило процесс стандартизации C++. Уместно заметить, что стандартизация C++ заняла значительно больше времени, чем это можно было предполагать. В процессе стандартизации к языку было добавлено много новых средств, а также внесено большое количество мелких изменений. В результате вариант C++, определенный комитетом ANSI/ISO, оказался значительно объемнее и сложнее первоначального замысла Страуструпа. Окончательный набросок стандарта был выпущен комитетом 14 ноября 1997 г., и весь стандарт стал действовать в 1998 г. Эту спецификацию C++ обычно называют стандартным C++.

## **Игра «Морской бой»**

Все это позволяет отобразить базовый функционал приложения (игра «Морской бой») достаточно просто и понятно для любого пользователя.

Правила игры «Морской бой» и ее история.

Сама же игра «Морской бой» является отличным примером для обучения разработке приложений на C++, благодаря своей простоте и возможности для фантазирования при программировании.

«Морской бой» — [игра](https://ru.wikipedia.org/wiki/%D0%98%D0%B3%D1%80%D0%B0) для двух участников, в которой игроки по очереди называют координаты на неизвестной им карте [соперника](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%BF%D0%B5%D1%80%D0%BD%D0%B8%D0%BA). Если у соперника по этим координатам имеется корабль (координаты заняты), то корабль или его часть «топится», а попавший получает право сделать ещё один [ход](https://ru.wikipedia.org/w/index.php?title=%D0%98%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D1%85%D0%BE%D0%B4&action=edit&redlink=1). Цель игрока — первым поразить все корабли противника.

Игра впервые была выпущена в виде [настольной игры](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D1%81%D1%82%D0%BE%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%B8%D0%B3%D1%80%D0%B0) компанией [Milton Bradley Company](https://ru.wikipedia.org/wiki/Milton_Bradley_Company" \o "Milton Bradley Company) в 1931 году.

Представленные ниже правила и нарушения относятся к бумажному варианту игры и немного отличаются от компьютерной версии по очевидным причинам.

**Правила размещения кораблей (флота).**

Игровое поле — квадрат 10×10 каждого игрока, на котором размещается флот кораблей.

Горизонтали обычно нумеруются сверху вниз, а вертикали помечаются буквами слева направо. При этом используются буквы русского алфавита от «а» до «к» (буквы «ё» и «й» обычно пропускаются), либо буквы латинского алфавита от «a» до «l». Иногда используется слово «республика» или «снегурочка», так как в этих 10-буквенных словах ни одна буква не повторяется. В данной реализации будут использоваться латинские буквы.

Размещаются:

* 1 корабль — ряд из 4 клеток («линкоры», или «четырёхпалубные»)
* 2 корабля — ряд из 3 клеток («крейсеры», или «трёхпалубные»)
* 3 корабля — ряд из 2 клеток («эсминцы», или «двухпалубные»)
* 4 корабля — 1 клетка («подлодки», или «однопалубные»)

Другой вариант названия: -трубные (напр. двухтрубный).

При размещении корабли не могут касаться друг друга углами (встречаются, однако, варианты, когда это не запрещается).

Редко (по договоренности) но, встречаются варианты игры, когда корабли могут размещаться в виде квадрата («четырехпалубные») или буквой «Г» («трех-» и «четырехпалубные»), но в нашем случае такое правило применятся не будет.

Рядом со «своим» квадратом чертится «чужой» такого же размера, только пустой. Это участок моря, где плавают чужие корабли противника.

При попадании в корабль противника — на чужом поле ставится крестик. Попавший стреляет ещё раз.

**Потопление кораблей противника**

Перед началом боевых действий игроки бросают жребий или договариваются, кто будет ходить первым.

Игрок, выполняющий ход, совершает выстрел — называет вслух [координаты](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D1%8F%D0%BC%D0%BE%D1%83%D0%B3%D0%BE%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%BE%D0%BE%D1%80%D0%B4%D0%B8%D0%BD%D0%B0%D1%82) клетки, в которой, по его мнению, находится корабль противника, например, «К1!» (верхняя правая клетка).

1. Если выстрел пришёлся в клетку, не занятую ни одним кораблём противника, то следует ответ «Мимо!» и стрелявший игрок ставит на чужом квадрате в этом месте точку. Право хода переходит к сопернику.
2. Если выстрел пришёлся в клетку, где находится многопалубный корабль (размером больше чем 1 клетка), то следует ответ «Ранил!» или «Попал!», кроме одного случая (см. пункт 3). Стрелявший игрок ставит на чужом поле в эту клетку крестик, а его противник ставит крестик на своём поле также в эту клетку. Стрелявший игрок получает право на ещё один выстрел.
3. Если выстрел пришёлся в клетку, где находится однотрубный корабль или последнюю непоражённую клетку многопалубного корабля, то следует ответ «Потоплен!» или «Убит!». Оба игрока отмечают потопленный корабль на листе. Стрелявший игрок получает право на ещё один выстрел.

Победителем считается тот, кто первым потопит все 10 кораблей противника. Проигравший имеет право попросить изучить после окончания игры у соперника игровое поле. Если проигравший находит какое-нибудь нарушение правил (см. ниже), то победа присуждается ему. Первоначальный победитель в свою очередь может попросить у соперника игровое поле для поиска нарушений. Если он их находит, то игроки сверяют чужие поля друг у друга. Если никаких несоответствий не замечено, то игра не засчитывается (оба проиграли). Тот, у кого чужое поле будет неверным (причём доказуемо неверным), будет проигравшим, а его соперник победителем. Игра также может закончиться раньше, чем когда будут потоплены все корабли, если нарушение правил будет замечено в течение игры. Проигравшим в этом случае будет считаться тот, у кого обнаружат нарушение правил, хотя он тоже может попросить игровое поле у соперника для поиска нарушений.

**Нарушения:**

Хотя следующие пункты и являются нарушениями, реализовать в данной работе третий пункт, а также один из подпунктов первого пункта, не будет представляться возможным, так как противником будет являться искусственный интеллект, а система координат будет задаваться изначально. Также, второй пункт будет относится только для бумажного варианта игры, в коде программы реализация такого пути нечестной игры не будет.

1. У игрока неправильно начерчено своё поле:
   * Количество кораблей не соответствует правилам;
   * Корабли касаются друг друга;
   * Неправильные размеры поля;
   * Неверная [система координат](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BA%D0%BE%D0%BE%D1%80%D0%B4%D0%B8%D0%BD%D0%B0%D1%82).
2. Игрок производил на своём игровом поле изменения, не предусмотренные правилами игры (в процессе игры можно ставить только точки и крестики и только по правилам), например, дорисовал недостающий корабль;
3. Игрок подглядел расположение кораблей противника;
4. Игрок [пропустил](https://ru.wikipedia.org/w/index.php?title=%D0%9F%D1%80%D0%BE%D0%BF%D1%83%D1%81%D0%BA_%D1%85%D0%BE%D0%B4%D0%B0&action=edit&redlink=1) свой ход.

# **Основная часть.**

## **1.1 Средства для выполнения задания.**

В качестве среды разработки приложения была выбрана Microsoft Visual Studio Ultimate 2013. Разработка шла под операционную систему Windows 8.1 (тестирование проводилось также на операционной системе Windows 7, прошедшее успешно, на других операционных системах тестирования не проводились).

## **1.2 Методы для выполнения задания.**

Использовались следующие стандартные библиотеки языка C++:

• <iostream>

iostream — [заголовочный файл](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B3%D0%BE%D0%BB%D0%BE%D0%B2%D0%BE%D1%87%D0%BD%D1%8B%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB) с классами, функциями и переменными для организации ввода-вывода в языке программирования [C++](https://ru.wikipedia.org/wiki/C%2B%2B). Он включён в [стандартную библиотеку C++](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_C%2B%2B). Название образовано от Input/Output Stream («поток ввода-вывода»). В языке C++ и его предшественнике, языке программирования [Си](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), нет встроенной поддержки ввода-вывода, вместо этого используется библиотека функций. iostream управляет вводом-выводом, как и [stdio.h](https://ru.wikipedia.org/wiki/Stdio.h" \o "Stdio.h) в Си. iostream использует [объекты](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) cin, cout, cerr и clog для передачи информации в и из [стандартных потоков](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D1%8B%D0%B5_%D0%BF%D0%BE%D1%82%D0%BE%D0%BA%D0%B8) ввода, вывода, ошибок (без буферизации) и ошибок (с буферизацией) соответственно. Являясь частью стандартной библиотеки C++, эти объекты также являются частью стандартного пространства имён — std.

Некоторые новые компиляторы (например, компилятор [Visual C++](https://ru.wikipedia.org/wiki/Visual_C%2B%2B" \o "Visual C++) из [Visual Studio .NET](https://ru.wikipedia.org/wiki/Visual_Studio_.NET) 2003) старые библиотеки <iostream.h> больше не поддерживают, вместо них нужно использовать только новые:#include <iostream>.

Пример программы с использованием библиотеки <iostream>:

#include <iostream>

int main ()

{

std::cout << "Example text" << std::endl;

return 0;

}

• <conio.h>

Сonio.h — [заголовочный файл](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B3%D0%BE%D0%BB%D0%BE%D0%B2%D0%BE%D1%87%D0%BD%D1%8B%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB), используемый в старых [компиляторах](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%82%D0%BE%D1%80), работающих в [операционных системах](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) [MS-DOS](https://ru.wikipedia.org/wiki/MS-DOS), для создания [текстового интерфейса пользователя](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F). Тем не менее, он не является частью [языка программирования Си](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%A1%D0%B8), [стандартной библиотеки языка Си](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_%D0%A1%D0%B8), [ISO C](https://ru.wikipedia.org/wiki/ISO_C) или требуемой стандартом [POSIX](https://ru.wikipedia.org/wiki/POSIX).

Этот заголовочный файл объявляет несколько библиотечных функций для работы с «консольным вводом и выводом» программы. Большинство компиляторов языка Си, предназначенных для DOS, Windows 3.x, Phar Lap, DOSX, [OS/2](https://ru.wikipedia.org/wiki/OS/2) или Win32 имели этот файл и обеспечивали сопутствующие библиотечные функции в библиотеке Си по умолчанию. Большинство компиляторов языка Си, предназначенных для [UNIX](https://ru.wikipedia.org/wiki/UNIX) и Linux, не имеют этого файла и не обеспечивают сопутствующих библиотечных функций.

Библиотечные функции, объявленные в conio.h, весьма различались в зависимости от компилятора. Первоначально реализованные в [Microsoft](https://ru.wikipedia.org/wiki/Microsoft" \o "Microsoft) [Visual C++](https://ru.wikipedia.org/wiki/Visual_C%2B%2B" \o "Visual C++) различные функции привязывались напрямую к нескольким первым функциям [DOS](https://ru.wikipedia.org/wiki/DOS), связанным с прерыванием 21h. Но библиотека, поставляемая с [Turbo C++](https://ru.wikipedia.org/w/index.php?title=Turbo_C%2B%2B&action=edit&redlink=1" \o "Turbo C++ (страница отсутствует)) и [Borland C++](https://ru.wikipedia.org/wiki/Borland_C%2B%2B" \o "Borland C++), не использует DOS API, а вместо этого напрямую обращается к видеопамяти для вывода информации и использования [сигналов прерывания BIOS](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB%D1%8B_%D0%BF%D1%80%D0%B5%D1%80%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_BIOS&action=edit&redlink=1). Компиляторы, предназначенные для операционных систем отличных от DOS, таких как [Linux](https://ru.wikipedia.org/wiki/Linux" \o "Linux), Windows и OS/2, обеспечивали различные реализации этих функций.

Данная библиотека используется исключительно для работы функции \_getch().

int \_getch(); Ждёт нажатия любой кнопки, и возвращает её код.

Пример программы с использованием библиотеки <conio.h>:

#include <conio.h>

int main ()

{

std::cout << "If press something key, "i" took his key number" << std::endl;

int I =\_getch();

return 0;

}

## **2.1 Интерфейс командной строки.**

Для реализации пошаговой игры «Морской бой» был использован текстовый интерфейс пользователя (в частности, интерфейс командной строки). Данное приложение рассчитано на игру между двумя людьми на одном компьютере. (см. Рис.1)

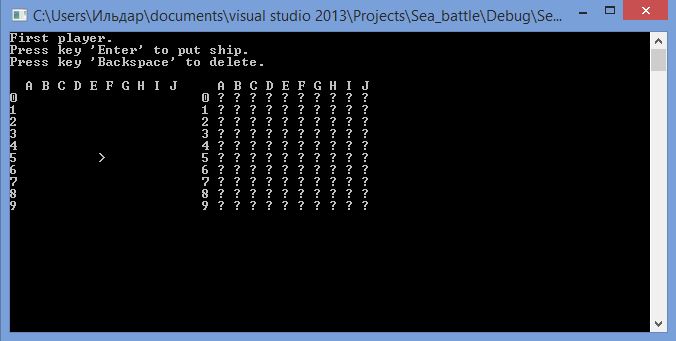


Рис.1

Данное приложение начинается с объявления игрока, который ходит первым. Соответственно, первым ходит первый игрок. Позже будет известно, что эта строка изменится, как только ход перейдет к следующему игроку. (см. Рис.2 и Рис.3)



Рис.2 Рис.3

Следующие две строки поясняют пользователю, какие клавиши нужно использовать, чтобы поставить корабль или его часть, или удалить его или его часть. Данные надписи показаны на протяжении всей игры, кроме момента перехода хода от одного игрока к другому. (см. Рис.4)



Рис.4

Игровое поле представляет из себя 2 квадрата. Каждый из квадратов для обоих игроков имеет размер 10х10, согласно правилам игры. Строки и столбцы соответственно подписаны цифрами и буквами латинского алфавита. Для визуального удобства столбцы были разделены пропусками, внешне схожими с пустыми столбцами, но на самом деле ими не являющимися. (см. Рис.5 и Рис.6)



Рис.5 Рис.6

Игрок управляет стрелкой, которая указывает на клетку квадрата, в которую может поставить пользователь блок корабля. (см. Рис.5)

Игрок работает в левом квадрате, т.к. правый будет использоваться в будущем для нахождения кораблей соперника. Аналогично для другого игрока.

Для того чтобы игрок поставил корабль ему нужно нажать клавишу Enter. Для удаления части корабля или всего однопалубного корабля игроку требуется нажать клавишу Backspace.

Корабли на левом квадрате игрового поля обозначаются символом "#". Пустые клетки обозначаются пробелом. На правом квадрате “неразведанные” клетки обозначаются"?", “разведанные” и подстреленные - "+", при промахе - "-", полностью уничтоженный корабль преобразовывается такими символами "X".(см. Рис.7)



Рис.7

Если игрок нарушает правило: ставит неправильное количество кораблей каждого типа; то высвечивается надпись « There is a wrong number of ships! Try again. ». (см. Рис.8)



Рис.8

После успешной расстановки кораблей ход переходит к следующему игроку. Об этом сигнализирует надпись, высвечивающая на экране. Все предыдущие пункты относятся и ко второму игроку. (см. Рис.9)



Рис.9

После расстановки кораблей обоими игроками ход переходит обратно к первому игроку. Идет следующий этап игры, когда противники пытаются «затопить» вражеские корабли путем поиска. Выбор игроков происходит на правом квадрате, в то время как на левом изображено расположение кораблей того же игрока, поставленных ранее. Игрок выбирает клетку, нажимает Enter и на клетке выводятся определенные символы, в случае попадания или ход переходит к следующему игроку при промахе. Происходит смена хода между пользователями.

Игра продолжается до того момента, как какой-нибудь из игроков не «потопит» все вражеские корабли. На экран выводится надпись о том, какой из игроков победил. (см. Рис.10)



Рис.10

# **Заключение**

Все поставленные задачи при выполнении курсовой работы были выполнены. После написания кода были выполнены отладка и проведено не одно тестирование на разных операционных системах. В исходном коде не выявлено ошибок, код успешно реализован. Игра «Морской бой» ясно отображается в командной строке, интерфейс командной строки смог достаточно ясно показать принцип работы игры, так, что любой пользователь может удостовериться в игре и опробовать ее.

Реализация данного кода помогла лучше узнать язык C++, укрепить, полученные ранее, общие знания о языке, закрепить материал об используемых функциях и библиотеках, присутствующих в данном коде.

# **Список литературы**

* Шилдт Герберт. C++ для начинающих. Серия «Шаг за шагом» / Шилдт Г.; пер. с англ. – М.:ЭКОМ Паблишерз, 2011 – 640 с.:ил.
* Франка П. С++: учебный курс . – СПБ.: Питер, 2003. – 521 с.:ил.
* Дэвис, Стефан, Р. C++ для "чайников", 4-е издание. : Пер. с англ. : — М. : Издательский дом '"Вильяме", 2003. — 336 с. : ил. : Парал. тит. англ.
* Шилдт Г. Самоучитель C++: Пер. с англ. — 3-е изд. — СПб.: БХВ-Петербург, 2003. — 688 с.

# **Приложение**

Важные составляющие исходного кода приложения «Морской бой»:

//Проверка верного расположения кораблей

bool

isValid()

{

int len[4] = { 0, 0, 0, 0 };

for (int i = 0; i < MAP\_SIZE; i++) {

if (map[player][i][0] == '#')

len[0]++;

else

len[0] = 0;

if (map[player][i][MAP\_SIZE - 1] == '#')

len[1]++;

else

len[1] = 0;

if (map[player][0][i] == '#')

len[2]++;

else

len[2] = 0;

if (map[player][MAP\_SIZE - 1][i] == '#')

len[3]++;

else

len[3] = 0;

if (len[0] > 4 || len[1] > 4 || len[2] > 4 || len[3] > 4)

return false;

}

for (int x = 1; x < MAP\_SIZE - 1; x++) {

for (int y = 1; y < MAP\_SIZE - 1; y++) {

if (map[player][y][x] == '#'){

int countX = 1;

int countY = 1;

if (map[player][y + 1][x - 1] == '#' ||

map[player][y - 1][x - 1] == '#' ||

map[player][y - 1][x + 1] == '#' ||

map[player][y + 1][x - 1] == '#')

return false;

int ix = x - 1;

while (ix >= 0 && map[player][y][ix--] == '#')

countX++;

ix = x + 1;

while (ix < MAP\_SIZE && map[player][y][ix++] == '#')

countX++;

int iy = y - 1;

while (iy >= 0 && map[player][iy--][x] == '#')

countY++;

iy = y + 1;

while (iy < MAP\_SIZE && map[player][iy++][x] == '#')

countY++;

if (countX != 1 && countY != 1)

return false;

if (countX > 4 || countY > 4)

return false;

}

}

}

return true;

}

//Проверка верного количества кораблей разного типа

bool

isValidShipsCount()

{

copy(player);

int limit[] = SHIP\_LIMIT;

for (int x = 0; x < MAP\_SIZE; x++) {

for (int y = 0; y < MAP\_SIZE; y++) {

if (map[MAP\_TEMP][y][x] == '#') {

map[MAP\_TEMP][y][x] = '+';

int countX = 0;

int countY = 0;

int ix;

int iy;

ix = x - 1;

while (ix >= 0 && map[MAP\_TEMP][y][ix] == '#') {

map[MAP\_TEMP][y][ix] = '+';

countX++;

ix--;

}

ix = x + 1;

while (ix < MAP\_SIZE && map[MAP\_TEMP][y][ix] == '#') {

map[MAP\_TEMP][y][ix] = '+';

countX++;

ix++;

}

iy = y - 1;

while (iy >= 0 && map[MAP\_TEMP][iy][x] == '#') {

map[MAP\_TEMP][iy][x] = '+';

countY++;

iy--;

}

iy = y + 1;

while (iy < MAP\_SIZE && map[MAP\_TEMP][iy][x] == '#') {

map[MAP\_TEMP][iy][x] = '+';

countY++;

iy++;

}

if (countY)

countX = countY;

if (limit[countX] == 0)

return false;

limit[countX]--;

}

}

}

return true;

}