МИНИСТЕРСТВО ОБРАЗОВАНИЯ ТВЕРСКОЙ ОБЛАСТИ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ «ТВЕРСКОЙ КОЛЛЕДЖ ИМЕНИ А.Н.КОНЯЕВА»

Лабораторный практикум

Графика в среде Turbo C++

Рекомендуется использовать для студентов 3-го курса
по дисциплине
"Основы программирования"
специальности 09.02.03

Тверь
2015

Содержание

Введение

Лабораторный практикум «Графика в среде Turbo C++» состоит из трёх лабораторных работ, предлагаемых к выполнению в определённом порядке с нарастанием сложности. Выполнение лабораторных работ ориентировано на среду программирования Turbo C++. Посредством выполнения предложенных лабораторных работ происходит ознакомление с приёмами начертания графических примитивов и приобретение навыков создания приложений с элементами мультипликации программным путём. Лабораторный практикум обеспечивает приобретение навыков, которые могут быть использованы при курсовом проектировании по дисциплине «Технология разработки программного продукта». Лабораторный практикум «Графика в среде Turbo C++» рассчитан на пользователей, владеющих навыками программирования в визуальной среде Turbo C++, и может быть предложен студентам 3, 4-х курсов специальности 09.02.03 « Программирование в компьютерных

ЛАБОРАТОРНАЯ РАБОТА №1

Дисциплина: Основы Программирования.

Тема:C++. Создание графических примитивов в среде Turbo C++.

Цель занятия: Научиться вычерчивать графические примитивы в среде Turbo C++.
Оборудование: ПК платформы IBM PC.

Теоретическое введение

Некоторые компоненты в Turbo C++имеют свойство Canvas(канва, холст), представляющие собой область компонента, на которой можно рисовать или отображать готовые изображения. Свойство Canvas используется для рисования пером (свойство **Pen**) и (кистью (свойство **Brush**) .

Канва (Canvas) обеспечивает:

- загрузку и хранение графических изображений;
- создание новых и изменение имеющихся изображений с помощью пера, кисти, шрифта;
- рисование и закраску различных фигур, линий, текстов;
- комбинирование различных изображений.

Каждая точка канвы имеет координаты X и У. Система координат канвы имеет началом левый верхний угол канвы. Координата X возрастает при перемещении слева направо, а координата У- при перемещении сверху вниз. Координаты измеряются в пикселах.

Пиксел – наименьший элемент поверхности рисунка. Важнейшее свойство пиксела- его цвет. Для описания цвета используется тип TColor.

Для того, чтобы вывести на поверхность объекта графический элемент, необходимо применить к свойству Canvas этого объекта соответствующий метод. Например, для вычерчивания на форме прямоугольника, можно записать:

```
Form1->Canvas->Rectangle(10,10,100,100);
```

Карандаш используется для вычерчивания точек, линий, контуров геометрических фигур: прямоугольников, окружностей, эллипсов, дуг и др. Вид линии, которую оставляет карандаш на поверхности холста, определяют свойства, представленные в Таблице1.

Таблица 1. - Свойства объекта Pen.

Color	Цвет линии
Width	Толщина линии
Style	Вид линии
Mode	Режим отображения

Например, указание

```
Form1->Canvas->Pen->Width:=2;  устанавливает толщину линии 2 пиксела.
Form1->Canvas->Pen->Color:=clRed;  - устанавливает красный цвет линии.
```

Style определяет стиль линии, который можно задать именованной константой. Список констант представлен в Таблице 2.

Таблица 2. - Список констант.

psSolid	Сплошная линия
psDash	Пунктирная линия, длинные штрихи
psDot	Пунктирная линия, короткие штрихи
psDashDot	Пунктирная линия, чередование длинного и короткого штрихов
psDashDot Dot	Пунктирная линия, чередование одного длинного и двух коротких штрихов
psClear	Линия не отображается

Кисть используется методами, обеспечивающими вычерчивание замкнутых областей и обладает двумя свойствами (Таблица 3).

Таблица 3. - Свойства кисти.

Color	Цвет закрашивания замкнутой области
Style	Стиль заполнения области

Константы, позволяющие задать стиль заполнения области, приведены в Таблице 4.

Таблица 4.- Константы задания стиля заполнения области.

bsSolid	Сплошная заливка
bsClear	Область не закрашивается
bsHorizontal	Горизонтальная штриховка
bsVertical	Вертикальная штриховка
bsFdiagonal	Диагональная штриховка с наклоном линий вперед
bsBDiagonal	Диагональная штриховка с наклоном линий назад
bsCross	Горизонтально-вертикальная штриховка в клетку
bsDiagCross	Диагональная штриховка, в клетку

Для вывода Текста на поверхности графического объекта используется метод **TextOut.**

Например, Form1->Canvas->TextOut(x,y,Текст) , где

x,y - координаты точки графической поверхности, от которой выполняется вывод текста.

Шрифт, который используется для вывода текста, определяется значением свойства Font.

Методы вычерчивания графических примитивов

Линия. Вычерчивание прямой линии осуществляет метод **LineTo**, инструкция вызова которого в общем виде выглядит следующим образом:

Компонент->**Canvas->LineTo(x,y);** .

Метод **LineTo** вычерчивает прямую линию от текущей позиции карандаша в точку с координатами, указанными при вызове метода.

Начальную точку линии можно задать, переместив карандаш в нужную точку графической поверхности при помощи метода **MoveTo(x,y).**

Окружность, эллипс можно начертить, используя метод **Ellipse**. Вызов метода выглядит: Компонент->**Canvas->Ellipse(x1,y1, x2,y2)** , где x1,y1, x2,y2 – координаты прямоугольника, внутри которого вычерчивается эллипс или, если прямоугольник является квадратом, окружность.

Дуга . Вычерчивание дуги выполняет метод Arc, инструкция вызова которого имеет вид: Компонент->**Canvas->Ellipse(x1,y1, x2,y2, x3,y3, x4,y4)** , где x1,y1, x2,y2, - параметры, определяющие эллипс (окружность), частью которого является вычерчиваемая дуга;

x3,y3 – параметры, определяющие начальную точку дуги;

x4,y4 - параметры, определяющие конечную точку дуги.

Прямоугольник вычерчивается методом **Rectangle,** вызов которого имеет вид:

Компонент->Canvas-> Rectangle(x1,y1,x2,y2); , где

x1,y1,x2,y2 – координаты левого верхнего и правого нижнего углов прямоугольника. Прямоугольник со скругленными углами вычерчивается методом:

RoungRec(x1,y1,x2,y2) .

Используемая литература

1. Юлий Кетков, Александр Кетков. Практика программирования: Visual Basic, C++ Builder, Delphi.”- СПб.:БХВ-Петербург, 2002.-464 с.: ил.
2.Никита Культин. Основы программирования в TURBO C++ (+ дистрибутив на CD). СПб.:БХВ-Петербург, 2013.-456с.: ил.

Контрольные вопросы

1. Какое свойство объекта позволяет рисовать или отображать на нем изображения?
2. Какие свойства Canvas используются для рисования?
3. С какого угла канвы начинается система координат?
4. Что представляет собой наименьший элемент поверхности рисунка?
5. Как называется свойство пиксела, обозначающее цвет?
6. Какие свойства имеет кисть?

Порядок выполнения

1. Выполнить задание 1;
2. Выполнить задание 2;
3. Выполнить задание 3;
4. Ответить на контрольные вопросы.

Задание 1. **Начертить графические примитивы: линию, прямоугольник, Эллипс** (Рисунок1).

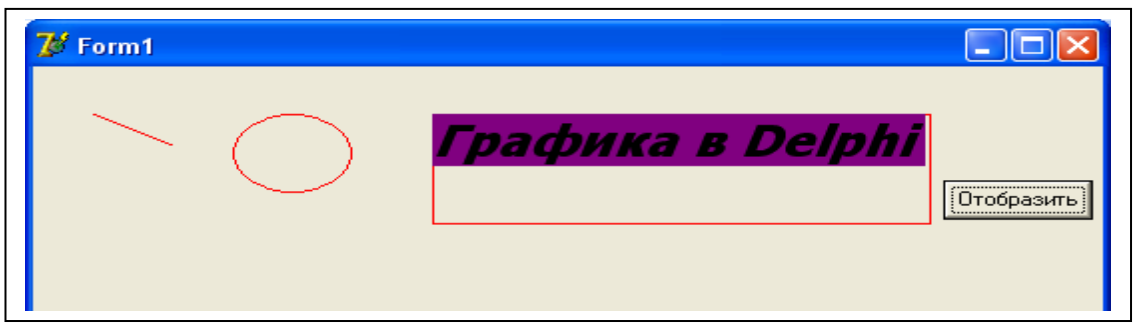


Рисунок 1

Задание 2. **Создать приложение, реализующее отображение геометрических объектов, изображенных на рисунке Рисунке 2.**

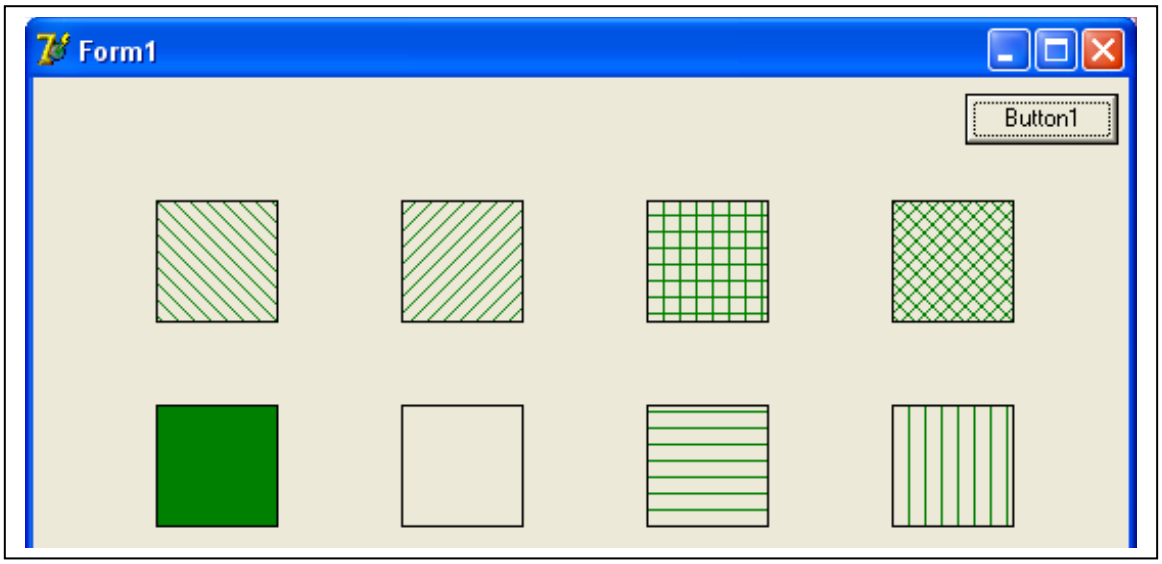


Рисунок 2

Оформление результата

Записать тему, цель занятия, приложить листинг исходной программы с результат работы. Ответить на контрольные вопросы.

Выставление оценки

Оценка «3» выставляется за выполнение Задания 1.
Оценка «4» выставляется за выполнение Заданий 1,2.
Оценка «5» выставляется за выполнение Заданий 1,2 (второе задание с использованием оператора цикла).

ЛАБОРАТОРНАЯ РАБОТА №2

Дисциплина: Основы программирования

Тема: C++. Создание приложений с мультипликацией в среде Turbo C++.

Цель занятия: приобретение навыков создания приложения с элементами мультипликации.

Оборудование: ПК платформы IBM PC.

Теоретическое введение

Под мультипликацией понимается движущийся и меняющийся рисунок. Рисунок может быть сформирован из графических примитивов (линий, окружностей, дуг, многоугольников и т. д. Обеспечить перемещение рисунка можно следующим образом: вывести рисунок на экран, затем через некоторое время стереть его или нарисовать цветом фона, а затем снова вывести этот же рисунок , но уже на некотором расстоянии от его первоначального расположения. Впечатление равномерного движения достигается путем подбора времени между выводом и удалением рисунка.

Для задания некоторого интервала времени можно использовать невизуальный компонент Timer (таймер) на вкладке System.Свойства компонента Timer перечислены в Таблице 1.

Таблица 1 - Свойства компонента Timer.

Свойство	**Определяет**
Name	Имя компонента,
Interval	Период генерации события OnTimer. Задается в миллисек.
Enabled	Разрешает (True) или запрещает(False) генерацию события OnTimer.

Компонент Timer является невизуальным и во время работы приложения не виден. Действия, которые необходимо выполнить по истечению интервала таймера, помещают в обработчик события OnTimer компонентаTimer.

Используемая литература

1. Никита Культин «Основы программирования в Turbo C++» (+ дистрибутив на CD). -СПб.: БХВ- Петербург, 2013. -456с.

Контрольные вопросы

1. Для чего служит компонент таймер?
2. С помощью какого свойства можно начать отсчет времени?
3. С помощью какого свойства можно задать период?

Порядок выполнения

1. Выполнить задание 1 по одному из вариантов;
2. Ответить на контрольные вопросы;
3. Составить отчетную карту.

Задание

Вариант 1. Отобразить в приложении работу светофора.

Вариант2. Отобразить на фоне звездного неба солнце с вращающейся вокруг него планетой.

Вариант 3. Отобразить на фоне звездного неба летящую летающую тарелку (Рисунок 1).

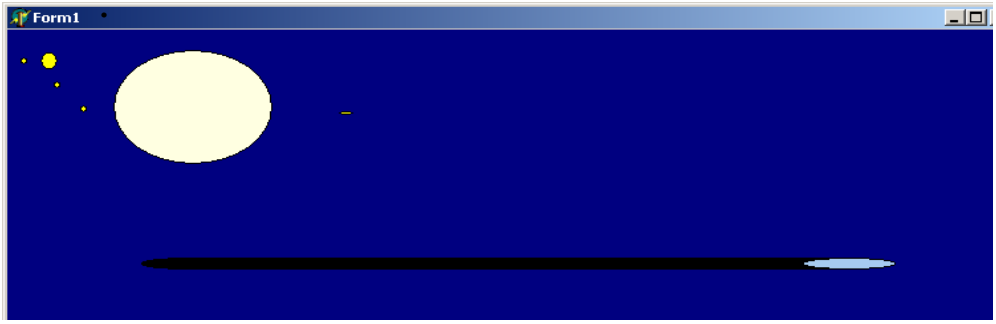
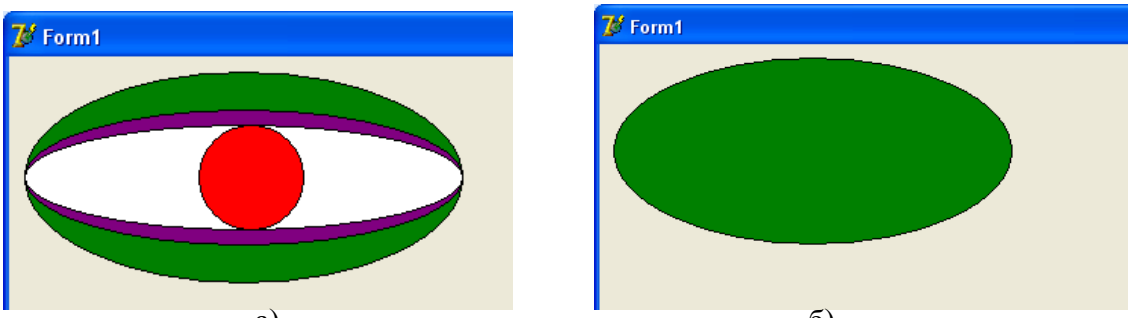


Рисунок 1.

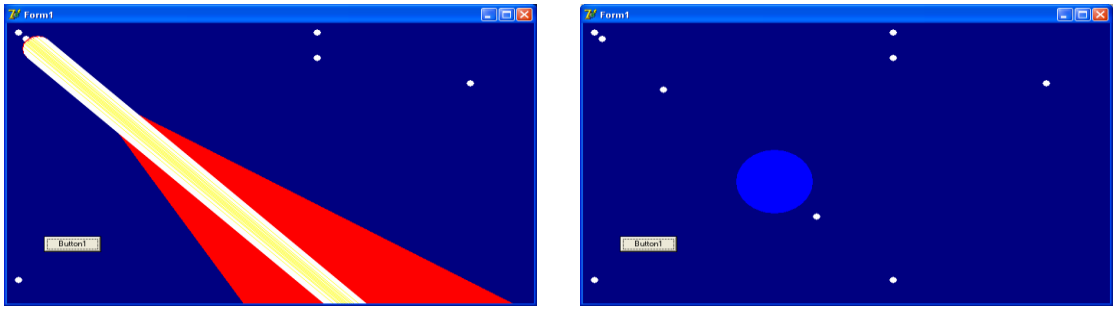
Вариант 4. Изобразить на форме космический глаз, периодически открывающийся и закрывающийся (Рисунок 2а,б)**.**



а) б)

Рисунок 2.

Вариант 5. Отобразить на фоне звездного неба периодически пролетающую комету (Рисунок 3а,б).



а) б)

Рисунок 3

Вариант 6. Отобразить плывущий по водной глади корабль Рисунок 4**.**

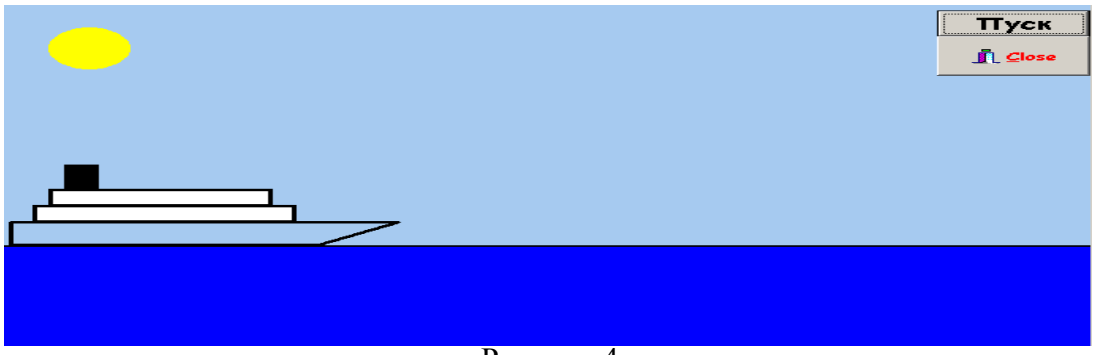
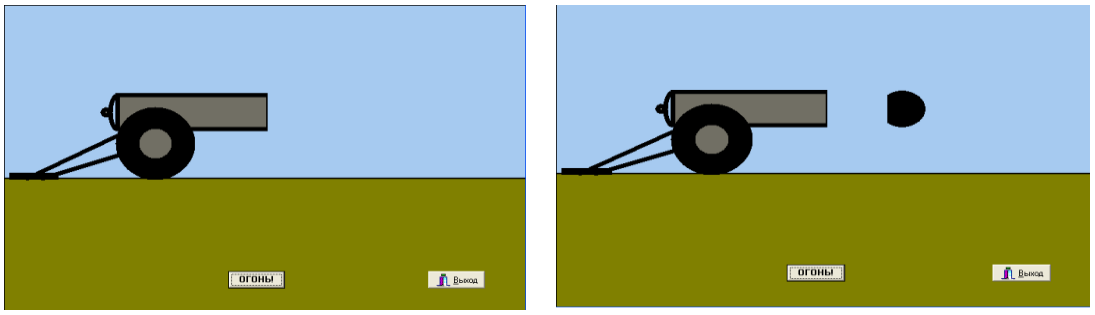


Рисунок 4.

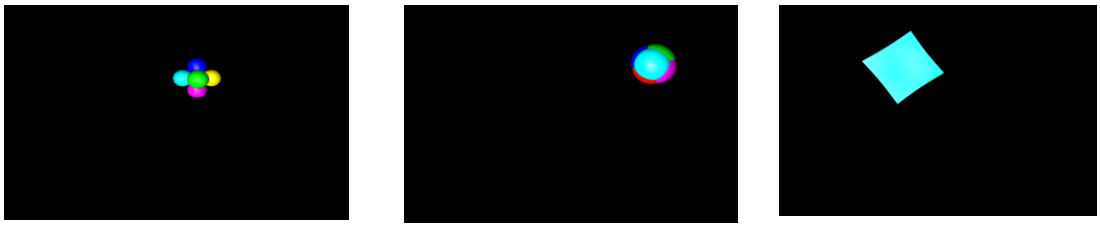
Вариант 7. Отобразить орудие, из которого через каждые 10 сек. вылетает снаряд (Рисунок 5а,б)**.**



а) б)

Рисунок 5.

Вариант 8. Создать заставку «Метаморфоза» (Рисунок 6а,б,с).



а) б) с)

Рисунок 6

Оформление результата

Записать тему, цель занятия, приложить листинг исходной программы с результат работы. Ответить на контрольные вопросы.

Выставление оценки

Оценка «3» выставляется за выполнение Задания 1.
Оценка «4» выставляется за выполнение Заданий 1,2.
Оценка «5» выставляется за выполнение Заданий 1,2 (второе задание с использованием оператора цикла).

ЛАБОРАТОРНАЯ РАБОТА №3

Дисциплина: Основы программирования

Тема: C++. Использование битовых образов в среде Turbo C++. Мультипликация.

Цель занятия: Приобрести навыки создания и использования битовых образов в среде Turbo C++, реализация перемещения одного сложного изображения на фоне другого.

Оборудование: ПК платформы IBM PC.

Теоретическое введение

Эффект перемещения картинки может быть создан путем периодической перерисовки картинки с некоторым смещением относительно ее прежнего положения. При этом предполагается, что перед выводом картинки в новой точке сначала удаляется предыдущее изображение. Удаление картинки может быть выполнено путем перерисовки всей фоновой картинки или только той ее части, которая перекрыта битовым образом движущегося объекта. Картинка может выводиться на канве применением метода *Draw* к свойству *Canvas* компонента (например, *Image*), а стирается путем копирования методом *CopyRect* нужной части фона из буфера на поверхность компонента (например, *Image*).

Для хранения битовых образов (картинок) фона и перемещающегося объекта, а также копии области фона, перекрываемой изображением объекта , используются объекты типа TBitMap. Программные объекты, используемые для копирования в буфер области фона, и на которую будет наложено изображение движущегося объекта и которая в последствии должна быть восстановлена из буфера, имеют тип Trect(прямоугольник). Для заполнения программных объектов типа Trect используется функция *Bounds(x,y,W,H*), где *x,y* - координата начальной точки, *W,H* – ширина и высота прямоугольной **области.**

Используемая литература

1. Никита Культин «Основы программирования в Turbo C++» (+ дистрибутив на CD). -СПб.: БХВ- Петербург, 2013. -456с.

Контрольные вопросы

1. С помощью какого свойства картинка может выводиться на канве?
2. Какие объекты используются для хранения битовых образов фона?
3. За счет чего достигается эффект передвижения картинки?

Порядок выполнения

1. Выполнить задание по одному из вариантов;
2. Ответить на контрольные вопросы;
3. Создать отчетную карту.

Задание. Создать фон и перемещаемый объект.
Реализовать перемещение объекта относительно фона.

Вариант 1. Фон- дорожный перекресток, объект- средство передвижения (Рисунок 1а,б)**.**

а) б)

Рисунок 1

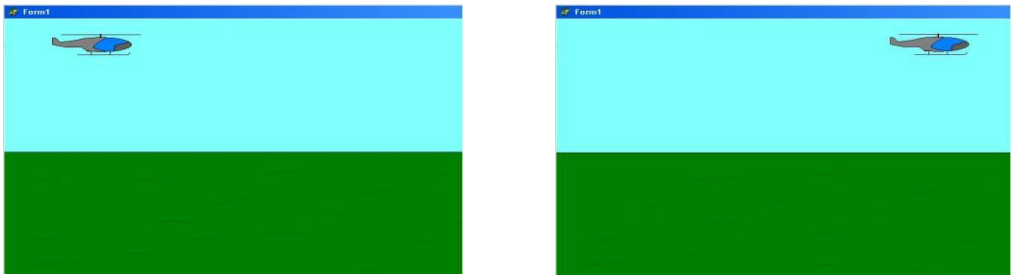
Вариант 2. Фон - луг, объект- трактор (Рисунок 2а,б).

а) б)

Рисунок 2

Вариант 3. Фон - горы, объект- альпинист.

Вариант 4. Фон - небо, объект – вертолет (Рисунок 3).



а) б)

Рисунок 3

Вариант 5. Фон - луг с кромкой леса, объект- велосипедист.

Вариант 6. Фон - подводная часть моря, объект – подводная лодка .

Вариант 7. Фон - луг, объект- трактор**.**

Вариант 2. Фон - небо, объект- парашютист**.**