

Министерство просвещения РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Глазовский государственный инженерно-педагогический университет имени В.Г. Короленко»

*Рассмотрено и утверждено на заседании кафедры
Математики и информатики
Протокол № 8 от 24.03.2025*

КОМПЛЕКТ ОЦЕНОЧНЫХ СРЕДСТВ
для проведения промежуточной аттестации в форме комплексного дифференцированного
зачета по междисциплинарному курсу

МДК 01.03 РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ
МДК 01.04 СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

специальность: 09.02.07. Информационные системы и программирование

квалификация: программист

Требования ФГОС к образовательным результатам:

В результате освоения дисциплины обучающийся должен уметь :	<ul style="list-style-type: none">- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;- создавать программу по разработанному алгоритму как отдельный модуль;- выполнять отладку и тестирование программы на уровне модуля;- осуществлять разработку кода программного модуля на современных языках программирования;- уметь выполнять оптимизацию и рефакторинг программного кода;- оформлять документацию на программные средства
В результате освоения дисциплины обучающийся должен знать :	<ul style="list-style-type: none">- основные этапы разработки программного обеспечения;- основные принципы технологии структурного и объектно-ориентированного программирования;- способы оптимизации и приемы рефакторинга;- основные принципы отладки и тестирования программных продуктов

Задания для проверки усвоения знаний.

Критерии оценки тестовых заданий.

Правильный ответ на вопрос оценивается в 1 балл, неправильный ответ или его отсутствие – ноль баллов.

Оценка	Процент правильных ответов
5(отлично)	90% - 100%
4(хорошо)	70% - 89%
3(удовлетворительно)	55% - 69%
2(неудовлетворительно)	54% и менее

Время на выполнение заданий: 1 академический час.

Уважаемый студент, Вам предлагается выполнить 21 задание в тестовой форме и 2 практических задания. Тест содержит задания на выбор одного верного ответа, нескольких ответов и установление соответствия. Время на выполнение всех заданий 90 минут.

Задания в тестовой форме

- Операция точки (операция доступа к члену класса) объединяет следующие два элемента (слева направо):
 - а) член класса и объект класса;
 - б) объект класса и класс;
 - в) класс и член этого класса;
 - г) объект класса и член этого класса.
- Классы полезны потому, что:
 - а) не занимают памяти, если не используются;
 - б) защищают свои данные от доступа со стороны других классов;
 - в) собирают вместе все аспекты, касающиеся отдельной вещи;
 - г) адекватно моделируют объекты реального мира.
- Какие отношения поддерживают объектно-ориентированные языки программирования?
 - а) ассоциация
 - б) наследование

- в) агрегация
- г) зависимость
- д) нет правильного ответа.

4. На каких принципах основывается объектно-ориентированное представление ПС?
- а) индивидуальностью;
 - б) состоянием;
 - в) поведением;
 - г) размером;
 - д) нет правильного ответа.
5. Какие пункты присущи структурному подходу при проектировании ПС?
- а) принцип структурирования данных;
 - б) принцип иерархического упорядочивания;
 - в) принцип непротиворечивости - заключается в обоснованности и согласованности элементов;
 - г) принцип формализации;
 - д) нет правильного ответа.
6. Какое сцепление имеют модули, если они имеют общие единицы, которые передаются от одного к другому как параметры, представляющие собой простые элементы данных?
- а) по образцу;
 - б) по данным;
 - в) по внешним ссылкам;
 - г) по кодам;
 - д) нет правильного ответа.
7. Какую связность имеет модуль, если общая структура данных является основой его организации как единого модуля?
- а) коммуникативная;
 - б) функциональная;
 - в) логическая;
 - г) последовательная;
 - д) нет правильного ответа.
8. Какую связность имеет модуль, выполняющий (реализующий) одну какую-либо определенную функцию?
- а) коммуникативная;
 - б) функциональная;
 - в) процедурная;
 - г) последовательная;
 - д) нет правильного ответа.
9. Какие интерфейсы могут разрабатываться при проектировании ПП?
- а) коммуникативная;
 - б) функциональная;
 - в) процедурная;
 - г) последовательная;
 - д) нет правильного ответа.
10. Какие цели преследуются при сборе бизнес требований?
- а) определение исходных данных;

- б) функциональное назначение программного продукта;
- в) характеристики качества, которые должны присутствовать в ПП;
- г) проблемы, которые нужно преодолеть;
- д) нет правильного ответа.

11. Какие типы комментариев в исходном коде являются обязательными?

- а) вводные;
- б) оглавления;
- в) пояснительные;
- г) описательные;
- д) нет правильного ответа.

12. Какое простейшее свойство характеризует ПО, приемлемостью величины погрешности в выдаваемых результатах?

- а) точность;
- б) адаптируемость;
- в) понятность;
- г) защищенность;
- д) нет правильного ответа.

13. Какой критерий качества характеризует ПО, позволяющее минимизировать усилия по внесению изменений и исправление ошибок?

- а) сопровождаемость;
- б) легкость применения;
- в) функциональность;
- г) эффективность;
- д) нет правильного ответа.

14. Какой критерий качества характеризует ПО безотказно выполнять заданные функции?

- а) надежность;
- б) легкость применения;
- в) функциональность;
- г) эффективность;
- д) нет правильного ответа.

15. Что является критерием качественного ПО?

- а) функциональность;
- б) надежность;
- в) эффективность;
- г) мобильность;
- д) нет правильного ответа.

16. Что учитывается при разработке ПО?

- а) периодичность использования ПО;
- б) количество пользователей;
- в) тип использования ПО;
- г) последствия отказов ПО;
- д) нет правильного ответа.

17. Какое ПО необходимо для написания новых программ?

- а) системное ПО;
- б) инструментальное ПО;
- в) прикладное ПО;

- г) любое ПО;
- д) нет правильного ответа.

18. Задание макроопределения в программе на языке C++ начинают с директивы препроцессора:
- а) `#ifdef`;
 - б) `#include`;
 - в) `#ifndef`;
 - г) `#define`;
 - д) `#endif`.
19. Какая команда препроцессора в программе на языке C++ служит для условной компиляции в случае когда некоторое макроопределение существует в настоящем программном модуле?
- а) `#ifdef`;
 - б) `#include`;
 - в) `#ifndef`;
 - г) `#define`;
 - д) `#endif`.
20. Какой знак в программе на языке C++ позволяет отличить директиву препроцессора от других команд?
- а) `->`
 - б) `.` (точка)
 - в) `%`
 - г) `#`
 - д) `$`
21. Блок операторов программного модуля, который подвергается условной компиляции, завершается директивой
- а) `#ifdef`;
 - б) `#include`;
 - в) `#ifndef`;
 - г) `#define`
 - д) `#endif`.

Практические задания

Задание 1. С помощью цветовой модели RGB задается оттенок цвета заливки прямоугольника по трем каналам: красному, зеленому и синему. Примените инкрементный метод конструирования программы для решения задачи. Программа на языке C++ с применением фреймворка Qt5. Одномодульная программа в файле `main.cpp`.

Задание 2. Примените инкрементный метод конструирования программы для решения следующей задачи.

Напишите программу на языке JavaScript с применением библиотеки API Yandex maps. Web-приложение выводит фрагмент карты, на котором отображает маршрут в форме зеленой линии из двух сегментов. В узлах сегментов на карте добавляются маркеры красного, зеленого и синего цветов. Координаты узлов сегментов заданы в исходном коде. На web-странице находятся 2 кнопки: Show Track и Hide Track, при нажатии которых соответственно отображается или скрывается линия, изображающая маршрут.

Дан пример последнего инкремента решения задачи. Требуется разработать последовательность инкрементов программы, которая приводит к данному образцу.

```

<!DOCTYPE html>
<html lang="ru">
<head>
  <title>Yandex map v0.1</title>
  <meta charset="utf-8">

  <style>
    html, body, #map {
      width: 100%;
      height: 100%;
      margin: 0;
      padding: 0;
    }

    #map {
      height: 50%;
    }
  </style>

  <script src="http://api-maps.yandex.ru/2.1.18/?lang=ru_RU&apikey=ad7bd5e9-
cdc2-47b7-8455-8d4800185ea7"></script>
  <script src="yamap01.js"></script>
</head>
<body>
<h2>Yandex map v0.1</h2>
<div id="map">
</div>
<a href="javascript:void(0)" class="btnCount">Start</a><br>
<a href="javascript:void(0)" class="btnCount">Stop</a><br>
<form action="">
  <NOBR>
    <input type="button" value="Show" onclick="onClickShow()">
    <input type="button" value="Hide" onclick="onClickHide()">
  </NOBR>
</form>
</body>
</html>

```

```

ymaps.ready(init);
var objPolyline1 = new Object();
var obj1 = new Object();
var mapobj1 = new Object();
function init() {
  console.log("Test ymaps: ready...");
  var map1;
  map1 = new ymaps.Map("map", {
    center: [55.7907909999963, 49.114689],

```

```

    zoom: 18
  });
  mapobj1 = map1;
  // https://tech.yandex.ru/maps/jsapi/doc/2.1/dg/concepts/geoobjects-docpage/
  var geoObject0 = new ymaps.GeoObject({
    geometry: {
      type: "Point",
      coordinates: [55.7907909999963, 49.114689]
    }
  });
  var placemark1Green = new ymaps.Placemark(
    [55.7906309999963, 49.115019],
    {}, { preset: "islands#greenIcon" }
  );
  obj1 = placemark1Green;
  var placemark2Red = new ymaps.Placemark(
    [55.7904659999963, 49.115370],
    {}, { preset: "islands#redIcon" }
  );
  map1.geoObjects.add(geoObject0);
  map1.geoObjects.add(placemark1Green);
  map1.geoObjects.add(placemark2Red);
  var polyline1 = new ymaps.Polyline([
    [55.7907909999963, 49.114689],
    [55.7906309999963, 49.115019],
    [55.7904659999963, 49.115370],
  ], {}, {
    strokeColor: "#00FF00",
    strokeWidth: 5,
  });
  objPolyline1 = polyline1;
  map1.geoObjects.add(polyline1);
}

```

```

function onClickShow() {
  console.log("Clicked button: Show" + obj1);
  mapobj1.geoObjects.add(obj1);
  mapobj1.geoObjects.add(objPolyline1);
  return false;
}

```

```

function onClickHide() {
  console.log("Clicked button: Hide" + mapobj1);
  mapobj1.geoObjects.remove(obj1);
  mapobj1.geoObjects.remove(objPolyline1);
  return false;
}

```

Критерии оценивания

Оценка 5 (отлично) ставится, если студент:

Набрал за тест 19-21 балл, практические задания выполнил без ошибок и недочетов либо допустил не более одного недочета.

Оценка 4 (хорошо) ставится, если студент:

Набрал за тест 15-18 баллов, практические задания выполнил полностью, но допустил в них не более одной негрубой ошибки и одного недочета, или не более двух недочетов.

Оценка 3 (удовлетворительно) ставится, если студент:

Набрал за тест 11-14 баллов, практические задания выполнил не менее чем на 50% или допустил не более двух грубых ошибок, или, при отсутствии ошибок, допустил четыре-пять недочетов.

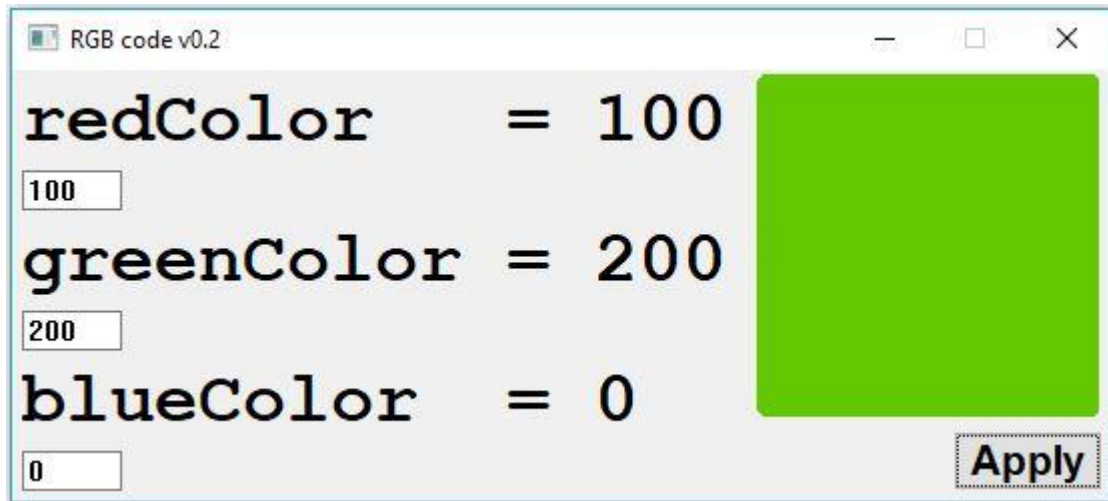
Оценка 2 (неудовлетворительно) ставится, если студент:

Набрал за тест менее 11 баллов, практические задания выполнил менее чем на 50%, или количество ошибок и недочетов превосходит норму, при которой может быть выставлена оценка 3 (удовлетворительно).

Эталон ответов

Практические задания

Задание 1 Вариант 2



```
// main.cpp
#include <QtWidgets>
```

```
QColor fillColor;
int redColor = 255;
int greenColor = 255;
int blueColor = 255;
QString strRedColor;
QString strGreenColor;
QString strBlueColor;
```

```
class MainWindow : public QMainWindow {
public:
    MainWindow();
    void paintEvent(QPaintEvent *event);
    void applyButtonClick();

private:
    QPushButton *applyButton;
    QLabel *redLabel;
    QLineEdit *redEdit;
    QLabel *greenLabel;
    QLineEdit *greenEdit;
    QLabel *blueLabel;
    QLineEdit *blueEdit;
};
```

```
MainWindow::MainWindow() {
    setWindowTitle("RGB code v0.2");
    setFixedSize(550, 215); // setFixedSize(550, 180);
    strRedColor = QString::number(redColor);
    strGreenColor.setNum(greenColor);
```

```

strBlueColor.setNum(blueColor);
fillColor.setRgb(redColor, greenColor, blueColor, 255);

QFont labelFont("Courier", 28, QFont::Bold);
QFont editFont("System", 12, QFont::Normal);
QFont buttonFont("Times", 16, QFont::Bold);
redLabel = new QLabel(this);
redLabel->setText("redColor = " + strRedColor);
redLabel->setGeometry(5, 5, 350, 40);
redLabel->setFont(labelFont);

greenLabel = new QLabel(this);
greenLabel->setGeometry(5, 75, 350, 40);
greenLabel->setFont(labelFont);

blueLabel = new QLabel(this);
blueLabel->setGeometry(5, 145, 350, 40);
blueLabel->setFont(labelFont);

redLabel ->setText("redColor  = " + strRedColor);
greenLabel->setText("greenColor = " + strGreenColor);
blueLabel ->setText("blueColor  = " + strBlueColor);

redEdit = new QLineEdit(this);
redEdit->setText(strRedColor);
redEdit->setGeometry(5, 50, 50, 20);
redEdit->setFont(editFont);
greenEdit = new QLineEdit(this);
greenEdit->setText(strGreenColor);
greenEdit->setGeometry(5, 120, 50, 20);
greenEdit->setFont(editFont);

blueEdit = new QLineEdit(this);
blueEdit->setText(strBlueColor);
blueEdit->setGeometry(5, 190, 50, 20);
blueEdit->setFont(editFont);

QPushButton *applyButton = new QPushButton("Apply", this);
applyButton->setGeometry(470, 180, 75, 30);
applyButton->setFont(buttonFont);
connect(applyButton, &QPushButton::clicked, this, &MainWindow::applyButtonClick);
}

void MainWindow::applyButtonClick()
{
    strRedColor  = redEdit->text();
    redColor     = strRedColor.toInt();
    strGreenColor = greenEdit->text();
    greenColor   = strGreenColor.toInt();
    strBlueColor = blueEdit->text();
    blueColor    = strBlueColor.toInt();
    redLabel ->setText("redColor  = " + strRedColor);
    greenLabel->setText("greenColor = " + strGreenColor);
    blueLabel ->setText("blueColor  = " + strBlueColor);
}

```

```

fillColor.setRgb(redColor, greenColor, blueColor, 255);
update();
}

void MainWindow::paintEvent(QPaintEvent *)
{
    QPainter painter(this);
    // fillColor.fromRgb(100, 100, 100, 255);
    painter.setPen(QPen(fillColor, 6, Qt::SolidLine, Qt::RoundCap));
    // painter.setBrush(QBrush(fillColor, Qt::SolidPattern));
    painter.setBrush(fillColor);
    painter.drawRect(375, 5, 165, 165); //(340, 5, 200, 135);
}

void MainWindow::paintEvent(QPaintEvent *)
{
    QPainter painter(this);
    // fillColor.fromRgb(100, 100, 100, 255);
    painter.setPen(QPen(fillColor, 6, Qt::SolidLine, Qt::RoundCap));
    // painter.setBrush(QBrush(fillColor, Qt::SolidPattern));
    painter.setBrush(fillColor);
    painter.drawRect(340, 5, 200, 135);
}

int main(int argc, char *argv[]) {
    QApplication app(argc, argv);
    MainWindow mainWin;
    mainWin.show();
    return app.exec();
}
Folder:
rgb02
  rgb02.pro
  main.cpp
// readme.txt
// Compilation Qt5 project
qmake rgb02.pro
mingw32-make
// rgb02.pro
QT += widgets
SOURCES += main.cpp

```

Задание 2

```

<!DOCTYPE html>
<html lang="ru">
<head>
  <title>Yandex map v0.1</title>
  <meta charset="utf-8">
  <style>
    html, body, #map {
      width: 100%;
      height: 100%;
      margin: 0;
    }
  </style>

```

```

        padding: 0;
    }
    #map {
        width: 70%;
        height: 30%;
    }
</style>
<script src="http://api-maps.yandex.ru/2.1.18/?lang=ru_RU&apikey=ad7bd5e9-cdc2-47b7-8455-8d4800185ea7">
</script>
<script src="yamax01.js"></script>
</head>
<body>
<h2>Yandex map v0.1</h2>
<div id="map"></div>
<form action="">
<NOBR>
    <input type="button" value="Show Track" onclick="onClickShowTrack()">
    <input type="button" value="HideTrack" onclick="onClickHideTrack()">
</NOBR>
</form>
</body>
</html>

```

Инкремент 1

```
ymaps.ready(init);
```

```
function init() {
    console.log("Test ymaps: ready...");
}
```

Инкремент 2

```
ymaps.ready(init);
```

```
function init() {
    console.log("Test ymaps: ready...");
    var myMap;
    myMap = new ymaps.Map("map", {
        center: [55.79, 49.10],
        zoom: 18
    });
}
```

Инкремент 3

```
ymaps.ready(init);
```

```
function init() {
    console.log("Test ymaps: ready...");
    var myMap;
    myMap = new ymaps.Map("map", {
        center: [55.79, 49.10],
        zoom: 18
    });
}
```

```
// https://tech.yandex.ru/maps/jsapi/doc/2.1/dg/concepts/geoobjects-docpage/
var myGeoObject = new ymaps.GeoObject({
  geometry: {
    type: "Point", // тип геометрии - точка
    coordinates: [55.79, 49.101] // координаты точки
  }
});
var myPlacemark = new ymaps.Placemark([55.79, 49.099]);
myMap.geoObjects.add(myPlacemark);
myMap.geoObjects.add(myGeoObject);
}
```

Инкремент 4

ymaps.ready(init);

```
function init() {
  console.log("Test ymaps: ready...");
  var map1;
  map1 = new ymaps.Map("map", {
    center: [55.79079099999963, 49.114689],
    zoom: 18
  });
  // https://tech.yandex.ru/maps/jsapi/doc/2.1/dg/concepts/geoobjects-docpage/
  var geoObject0 = new ymaps.GeoObject({
    geometry: {
      type: "Point",
      coordinates: [55.79079099999963, 49.114689]
    }
  });
  var placemark1Green = new ymaps.Placemark(
    [55.79063099999963, 49.115019],
    {}, { preset: "islands#greenIcon" }
  );
  var placemark2Red = new ymaps.Placemark(
    [55.79046599999963, 49.115370],
    {}, { preset: "islands#redIcon" }
  );
  map1.geoObjects.add(geoObject0);
  map1.geoObjects.add(placemark1Green);
  map1.geoObjects.add(placemark2Red);
  var polyline1 = new ymaps.Polyline([
    [55.79079099999963, 49.114689],
    [55.79063099999963, 49.115019],
    [55.79046599999963, 49.115370],
  ], {}, {
    strokeColor: "#00FF00",
    strokeWidth: 5,
  });
  map1.geoObjects.add(polyline1);
}
```

Инкремент 5

<!DOCTYPE html>

```

<html lang="ru">
<head>
  <title>Yandex map v0.1</title>
  <meta charset="utf-8">

  <style>
    html, body, #map {
      width: 100%;
      height: 100%;
      margin: 0;
      padding: 0;
    }

    #map {
      height: 50%;
    }
  </style>

  <script src="http://api-maps.yandex.ru/2.1.18/?lang=ru_RU&apikey=ad7bd5e9-cdc2-47b7-8455-8d4800185ea7"></script>
  <script src="yamap01.js"></script>
</head>
<body>
<h2>Yandex map v0.1</h2>
<div id="map">
</div>
<a href="javascript:void(0)" class="btnCount">Start</a><br>
<a href="javascript:void(0)" class="btnCount">Stop</a><br>
<form action="">
  <NOBR>
    <input type="button" value="Show" onclick="onClickShow()">
    <input type="button" value="Hide" onclick="onClickHide()">
  </NOBR>
</form>
</body>
</html>

```

```

ymaps.ready(init);
var objPolyline1 = new Object();
var obj1 = new Object();
var mapobj1 = new Object();
function init() {
  console.log("Test ymaps: ready...");
  var map1;
  map1 = new ymaps.Map("map", {
    center: [55.7907909999963, 49.114689],
    zoom: 18
  });
  mapobj1 = map1;
// https://tech.yandex.ru/maps/jsapi/doc/2.1/dg/concepts/geoobjects-docpage/
var geoObject0 = new ymaps.GeoObject({
  geometry: {
    type: "Point",
    coordinates: [55.7907909999963, 49.114689]
  }
});

```

```

    }
  });
  var placemark1Green = new ymaps.Placemark(
    [55.79063099999963, 49.115019],
    {}, { preset: "islands#greenIcon" }
  );
  obj1 = placemark1Green;
  var placemark2Red = new ymaps.Placemark(
    [55.79046599999963, 49.115370],
    {}, { preset: "islands#redIcon" }
  );
  map1.geoObjects.add(geoObject0);
  map1.geoObjects.add(placemark1Green);
  map1.geoObjects.add(placemark2Red);
  var polyline1 = new ymaps.Polyline([
    [55.79079099999963, 49.114689],
    [55.79063099999963, 49.115019],
    [55.79046599999963, 49.115370],
  ], {}, {
    strokeColor: "#00FF00",
    strokeWidth: 5,
  });
  objPolyline1 = polyline1;
  map1.geoObjects.add(polyline1);
}

```

```

function onClickShow() {
  console.log("Clicked button: Show" + obj1);
  mapobj1.geoObjects.add(obj1);
  mapobj1.geoObjects.add(objPolyline1);
  return false;
}

```

```

function onClickHide() {
  console.log("Clicked button: Hide" + mapobj1);
  mapobj1.geoObjects.remove(obj1);
  mapobj1.geoObjects.remove(objPolyline1);
  return false;
}

```