**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ**

**ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ**

**им. проф. М.А. Бонч-Бруевича»**

**(СПбГУТ)**

**Санкт-Петербургский колледж телекоммуникаций**

**Отчет о выполнении  
практического занятия №14-15**

Выполнил: Обучающиеся 4 курса, 581 группы,

Филипович Валерий Анатольевич

Состав бригады: -

Проверил: преподаватель  
Баталов Дмитрий Иннокентьевич

Санкт-Петербург

2021 г.

**Постановка задачи**

Реализовать два консольных приложения со следующими условиями:

* **Задание 1.** Запустить на выполнение код программы DrawCanvas.java примитивного графического редактора, приведенный в конспекте лекционного занятия «Занятие № 12. Программирование графики и обработка событий».

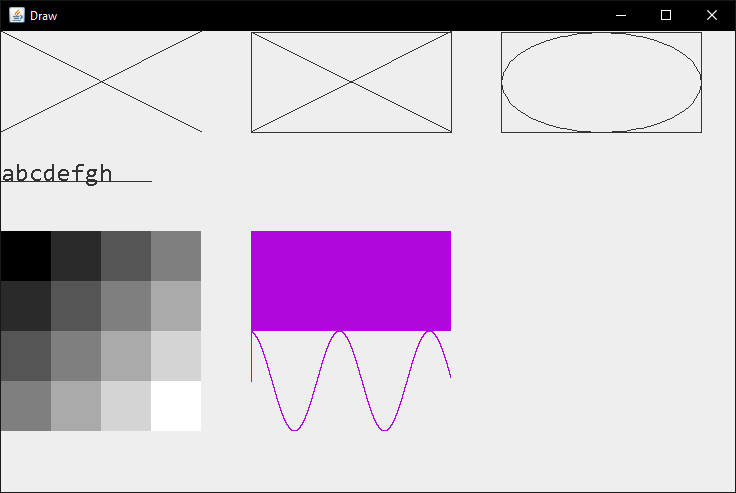
Далее на основе программы DrawCanvas.java выполнить отображение всех графических элементов, приведенных в конспекте лекционного занятия «Занятие № 12. Программирование графики и обработка событий». Для этого нужно вносить изменения в код программы для того, чтобы получить правильный результат. Обращайте внимание на изменение в работе программы при нажатии мыши над окном программы. Научитесь изменять размер окна программы, элементов в окне программы, шрифт и т. д. Добейтесь, чтобы окно программы закрывалось при нажатии Х в правом верхнем углу окна программы.

* **Задание 2.** Изучить справочную информацию, данную ниже. Выполнить код программы данный в Примере 1. «Выпадающее меню». Добиться понимания назначения элементов программы.

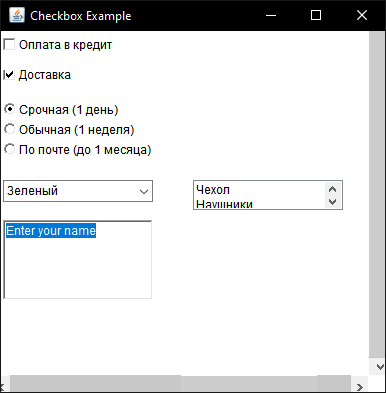
**Ход выполнения работы**

1. Создал новый консольный проект «PZ\_14-15»;
2. Добавил в код фрагмент, в котором создается графическое окно JFrame, в котором представлены все графические элементы, описанные в первой части лекционного занятия «Занятие № 12. Программирование графики и обработка событий»;
3. Добавил в код фрагмент, в котором создается графическое окно Frame, в котором представлены все отдельные контролы (элементы управления), описанные во второй части лекционного занятия «Занятие № 12. Программирование графики и обработка событий»;
4. Добавил в код фрагмент отдельного окна Frame с использованием класса FlowLayout;
5. Добавил в код фрагмент отдельного окна Frame с использованием класса BorderLayout;
6. Добавил в код фрагмент отдельного окна Frame с использованием класса GridLayout;
7. Добился успешного закрывания программ при нажатии X в правом верхнем углу окна программы.
8. Создал новый консольный проект «PopupMenu»;
9. Добавил в код фрагмент, описанный в документе «МДК.03.02\_ИСРПО\_Тема2.1\_ПЗ\_14-15», в котором реализовано окно, при нажатии на которое в области апплета появляется меню для выбора цвета, и область апплета перекрашивается;

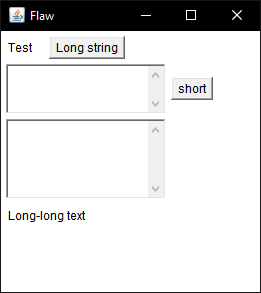
**Результат выполнения программы**



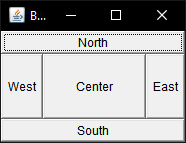
*Рис. 1. Фрейм графических элементов*



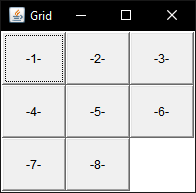
*Рис. 2. Фрейм c контролами*



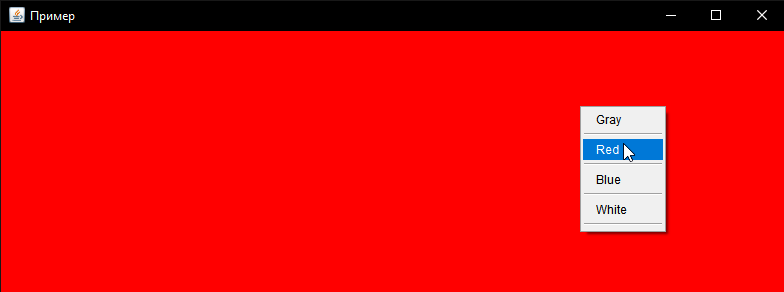
*Рис. 3. Фрейм с использованием класса* *FlowLayout*



*Рис. 4. Фрейм с использованием класса BorderLayout*



*Рис. 5. Фрейм с использованием класса GridLayout*



*Рис. 6. Результат работы программы «PopupMenu»*

**Ссылка на GitHub**

<https://github.com/ValeriyFilipovich/SPbCT_FilipovichVA/tree/master/Practice_tasks/>

**Ответы на контрольные вопросы**

1. ***Что такое GUI?***

Графический интерфейс пользователя.

1. ***Какие два типа компонентов GUI существует?***

Abstract Windows Toolkit (AWT) и Java Foundation Classes (JFC)

1. ***Что такое элементы управления и что такое контейнеры?***

Элементы управления – то, с помощью чего пользователь может управлять программой. Чаще всего представляют собой графические контролы в оконном приложении.

Контейнеры — это объекты, которые содержат компоненты.

1. ***Какие классы элементов управления существуют?***

Button, Checkbox, CheckboxGroup, Choice, List, Scrollbar, Label, TextComponent (TextArea и TextField)…

1. ***Что необходимо сделать, чтобы ввести компонент в контейнер?***

Необходимо в методе класса Container (или его подкласса) просто создать экземпляр объекта, дать ему имя и добавить его в контейнер (с помощью метода add());

1. ***Как можно перехватить и обработать события, пришедшие от компонентов?***

Классы, производные от Window, обрабатывают события WindowEvent. Для их прослушивания используется интерфейс WindowListener. Интерфейс включает несколько методов. Чтобы обрабатывать одно из событий лучше использовать адаптер - WindowAdapter.

1. ***Какие типы переключателей существуют?***

Неисключающие флажки (переключатели с независимой фиксацией) могут быть выбраны независимо от состояния других флажков. Такие флажки являются самостоятельными объектами класса Checkbox, для их создания используются конструкторы Checkbox() и Checkbox(String label).

Исключающие флажки (переключатели с зависимой фиксацией) позволяют одновременно выбрать только один элемент в группе флажков. Если выбирается другой элемент в группе, то выбор предыдущего флажка отменяется, а новый выбор выделяется.

1. ***Как несколько переключателей объединить в группу?***

Для создания группы переключателей сначала создается объект класса CheckboxGroup, а затем создаются объекты класса Checkboх, входящие в эту группу, для чего используется конструктор Checkbox(String label, CheckboxGroup group, boolean state).

1. ***Чем отличаются выпадающие и раскрывающиеся списки? Как осуществляется в них выбор элементов?***

Применяя меню выбора (класс Choice), можно выбирать только один элемент - множественный выбор не разрешается. При необходимости множественного выбора придется воспользоваться раскрывающимся списком.

1. ***При помощи какого метода обрабатываются сообщения от линейки прокрутки?***

При использовании прокрутки вырабатывается событие класса AdjustmentEvent, для его прослушивания применяется интерфейс AdjustmentListener, в котором предусмотрен один метод - adjustmentValueChanged().

1. ***Порождают ли события компоненты класса Label и для чего используются эти компоненты? В чем преимущества их использования перед обычным выводом текста методами класса Graphics?***

Они не порождают событий, они используются просто для размещения на экране строк текста. Преимущество использования этих элементов управления для вывода текста перед выводом текста методом drawString() класса Graphics заключается в том, что эти элементы управления (вместе с их содержимым) имеют способность перемещаться по экрану всякий раз, когда изменяется размер контейнера, содержащего эти метки.

1. ***Что такое текстовые поля и текстовые области? Чем они отличаются?***

Поля редактирования типа TextArea может использоваться как для вывода, так и для ввода и редактирования текста. Поля редактирования этого типа состоят из нескольких строк текста и имеют полосы прокрутки. Напротив, поля редактирования типа TextField состоят из одной строки и не имеют полос прокрутки.

1. ***Что такое контейнеры? Какие основные виды контейнеров существует?***

Контейнеры — это объекты (компоненты), позволяющие помещать в себя другие различные компоненты. Класс контейнеров Сontainer - подкласс класса Component. Существует два вида основных вида контейнеров: панели (класс Panel, подклассом которого является класс Applet) и окна (класс Window, подклассами которого являются Frame и Dialog.

1. ***Для чего чаще всего используются панели?***

Панели очень часто используются в апплетах, имеющий сложный графический интерфейс пользователя.

1. ***Как можно рисовать в окне панели?***

В силу того, что апплеты фактически являются потомками класса Component, имеется возможность рисовать прямо в апплете, не прибегая для этого к специальным методам.

1. ***В чем основное отличие окон и панелей?***

Окна (класс Window), как и панели, являются общим классом контейнеров. Но в отличие от панелей окно Java представляет собой окно - объект операционной системы, существующий отдельно от окна WWW-броузера или программы просмотра апплетов.

Непосредственно класс Window никогда не используется, а используются три его подкласса Frame, Dialog и FileDialog.

1. ***Что является обязательным параметром конструктора при создании экземпляра класса окон?***

При создании окна обязательным параметром конструктора Window() является объект класса Frame.

1. ***Каковы отличительные особенности имеют фреймы?***

Оно автоматически может иметь главное меню (так как класс Frame реализует интерфейс MenuContainer), для него можно устанавливать форму курсора и пиктограмму. Внутри такого окна можно рисовать. Так как окно класса Frame произошло от класса Container, то в него можно добавлять различные компоненты и панели, так же как в делается для апплетов и панелей. Нужно отметить, что по умолчанию для окон класса Frame устанавливается режим размещения BorderLayout.

1. ***Какие методы должны быть переопределены в новом подклассе фреймов?***

Для создания подкласса класса Frame необходимо определить новый класс, унаследовав его от класса Frame.

После объявления класса можно создавать объекты этого класса, вызывая для отображения окна метод show().

1. ***Какой класс контейнеров автоматически поддерживает работу с меню? Почему он это делает автоматически?***

Окно автоматически может иметь главное меню (так как класс Frame реализует интерфейс MenuContainer), для него можно устанавливать форму курсора и пиктограмму. Внутри такого окна можно рисовать. Так как окно класса Frame произошло от класса Container, то в него можно добавлять различные компоненты и панели, так же как в делается для апплетов и панелей.

1. ***Как добавить меню в контейнер?***

Для того, что создать и добавить меню в контейнер, необходимо сначала создать панель меню (объект класса MenuBar), затем добавить в нее меню (объект класса Menu).

1. ***Как создать новое меню и добавить в него элементы?***

Для формирования главного меню создается объект класса MenuBar, а затем в него добавляются отдельные меню. После создания отдельных меню в них можно добавить элементы. Для этого нужно вызвать метод add() класса Menu.

Сформированные отдельные меню добавляются затем к панели меню при помощи метода add() класса MenuBar.

1. ***Какие существуют классы элементов меню?***

Класс MenuItem определяет поведение элементов меню. Пользуясь методами этого класса, можно блокировать или деблокировать отдельные элементы — это нужно делать, например, если в данный момент функция, соответствующая строке меню, недоступна или не определена. Можно также изменять текстовые строки, соответствующие элементам меню, что может пригодиться для переопределения их значения.

1. ***Какими методами обрабатываются события меню?***

Когда пользователь выбирает одну из команд меню, происходит генерация события класса ActionEvent. Для его обработки используется интерфейс ActionListener с одним методом actionPerformed. Для прослушивания команд меню можно добавить прослушивание к каждому пункту меню или же сразу ко всему меню

1. ***Для чего в основном используются окна диалогов?***

Диалоговые окна (класс Dialog) используются в основном для одноразового запроса информации у пользователя или для вывода небольших порций информации на экран.

1. ***Каковы важные отличия окон диалогов от фреймов?***

Диалоговые окна во всем подобны фреймам, но имеют два важных отличия: во-первых, они не являются реализацией интерфейса MenuContainer (а следовательно не реализуют панель меню автоматически); во-вторых, они могут иметь модальность - это значит, что можно сконструировать диалоговое окно, которое запретит пользователю обращаться к другим окнам (включая и окно WWW-навигатора) до тех пор, пока пользователь не произведет требуемого действия в этом диалоговом окне.

1. ***Объект какого класса должен обязательно быть родителем диалогового окна?***

Каждое диалоговое окно обязательно должно иметь фрейм в качестве родителя. Это значит, что диалоговое окно нельзя открыть непосредственно из апплета. Чтобы создать диалоговое окно, необходимо сначала завести фрейм, даже если его единственным значением будет служить родителем для диалогового окна. Только если апплет уже использует фреймы, можно обойтись без этой подготовительной стадии.

1. ***Как создать диалог своего класса?***

Для того, чтобы создать свою диалоговую панель, необходимо определить новый класс, унаследовав его от класса Dialog. Затем можно создавать диалоги этого класса следующим оператором:

MessageBox msgBox=new MessageBox("String of message",new Frame(),

"Message box",true)

1. ***Для чего предназначены менеджеры компоновки? Какие существуют режимы размещения?***

Способ размещения компонентов в контейнере зависит от менеджера компоновки, ассоциированного с этим контейнером. С помощью этих менеджеров можно легко и быстро обеспечить необходимое расположение компонентов относительно друг друга и включающего их контейнера.

Режимы: FlowLayout, GridLayout, BorderLayout, CardLayout, GridBagLayout.