## Практические задания

Задание лабораторной работы делятся на обязательные и дополнительные. Дополнительные задания принимаются только после выполнения и защиты хотя бы одного обязательного задания.

Работа рассчитана на индивидуальное выполнение в течение одной недели.

Для получения оценки «**удовлетворительно**» необходимо выполнить хотя бы **часть** обязательных заданий, набрав при этом **не менее 50 баллов**.

Для получения оценки «**хорошо**» необходимо выполнить хотя бы **часть** обязательных заданий и, возможно, часть дополнительных, набрав при этом **не менее 150 баллов**.

Для получения оценки «**отлично**» необходимо выполнить **часть** обязательных заданий, и, возможно, часть дополнительных, набрав при этом не менее **250 баллов**.

### Обязательные задания

#### Задание 1. 30 баллов

В этом задании для визуализации разрешается использовать произвольные графическое API и UI-framework. Допускается использовать и веб-технологии (HTML, CSS, JavaScript).

Разработайте GUI-приложение, отображающее при помощи векторных примитивов (прямоугольники, многоугольники и другие) Ваши инициалы разноцветными буквами.

Пример визуализации инициалов студента Петра Ильича Чайковского:

##### Бонус 20 баллов за создание анимированного изображения

Доработайте приложение так, чтобы буквы Ваших инициалов подпрыгивали (равноускоренное движение). Фаза прыжка каждой буквы должна несколько отличаться от фазы прыжка предыдущей.

На возможное мерцание букв пока не обращайте внимание.

#### Задание 2 – 60 баллов

В этом задании для визуализации разрешается использовать произвольное графическое API. Разрешается использовать и веб-технологии (HTML, CSS, JavaScript).

Разработайте GUI-приложение графическое приложение, визуализирующее с использованием векторных примитивов картинку на одну из следующих тем:

* **Домик в деревне**. Домик с крышей, трубой, окном, дверью и забором.
* **Троллейбус**. Троллейбус с несколькими окнами, колесами, а также токоприемниками, соединенными с проводами.
* **Один из героев мультсериала South Park.** Постараться соблюсти визуальное сходство с прототипом
* **Системный блок компьютера и монитор (вид спереди).** У системного блока должен быть виден CD-ROM, кнопка включения и цветные индикаторы (питание, жесткий диск).
* **Цветок**. У цветка должно быть несколько листьев и лепестков, а также стебель.
* **Грузовик (вид сбоку)**. Кузов, колеса, окна, дверь.
* **Космическая ракета**.
* **Тема по выбору студента**. Тема, предложенная студентом не должна уступать по сложности предложенным выше вариантам.

##### Бонус 30 баллов за возможность таскания объектов при помощи мыши

Пользователь должен иметь возможность «таскать» картинку при помощи мыши методом drag-and-drop.

### Дополнительные задания

#### Задание 3

##### Вариант 1. 30 баллов.

Ознакомьтесь с существующими алгоритмами построения отрезков прямых и разработайте графическое приложение, выполняющее визуализацию отрезка прямой заданного цвета между двумя произвольными точками экрана. Построение отрезка должно выполняться вручную (точка за точкой), без использования встроенных функций используемого графического API вроде LineTo.

Отрезок задается координатами начала и конца, которые могут выходить за пределы области экрана. Координаты концов отрезка могут выходить за пределы экрана. Программа должна эффективно отсекать части отрезка, выходящие за пределы окна.

##### Вариант 2. 50 баллов.

Ознакомьтесь с существующими алгоритмами построения окружностей и разработайте графическое приложение, выполняющее визуализацию окружности заданного цвета. Построение окружности должно выполняться вручную (точка за точкой) без использования встроенных функций используемого графического API, таких как Ellipse.

Окружность задается координатами центра и радиусом, любая часть окружности может выходить за пределы области экрана. Программа должна эффективно отсекать части окружности, выходящие за пределы экрана (окна).

###### Бонус до 20 баллов за возможность закрашивания внутренней части окружности

Бонус начисляется за возможность закрашивания внутренностей окружности цветом, задаваемым отдельно от цвета самой окружности.

#### Задание 4. 150 баллов

Разработайте с использованием произвольного графического API и UI-фреймворка компьютерную игру «Виселица».

Суть игры заключается в том, что компьютер загадывает слово и дает игроку некоторую подсказку. Игрок за ограниченное количество попыток должен угадать это слово, называя его по буквам. За каждую неверно названную букву игрок лишается попытки, что графически изображается рисованием новой части тела «висельника» (веревка, голова, тело, нога, нога, рука, рука). При правильно угаданной букве показываются все ее вхождения внутри слова. Игра заканчивается либо победой игрока при угадывании всех букв слова, либо его проигрышем при исчерпании имеющегося количества попыток. Примерный вид экрана программы показан на рисунке:

Выбор букв осуществляется игроком при помощи мыши. По окончании игры пользователю показывается сообщение с информацией о выигрыше/проигрыше с предложением начать новую игру. При утвердительном ответе программа должна предложить пользователю новое слово, в противном случае программа завершить свою работу.

Правильно угаданные буквы выделяются зеленым цветом, а неправильно угаданные – красным. Программа не должна позволять повторный выбор букв (при их выборе ничего не должно происходить).

Слова и задания должны загружаться программой из внешнего текстового файла (необходимо выбрать подходящий формат для их хранения) и предлагаться пользователю случайным образом.

##### Бонус до 50 баллов за возможность переключения визуального представления игры

Бонус начисляется за возможность в любой момент переключаться между основным и альтернативным вариантами визуального представления игрового при помощи меню. Состояние игры при этом изменяться не должно. Пример альтернативного способа визуализации

