

**Разработка технического задания к программному продукту R Tutorial**

*Дмитриева Виктория Александровна*

*Хакасский государственный университет им. Н.Ф. Катанова  
магистрант*

*Григорьева София Сергеевна*

*Хакасский государственный университет им. Н.Ф. Катанова  
магистрант*

*Научный руководитель:*

*Голубничий Артем Александрович*

*Хакасский государственный университет им. Н.Ф. Катанова  
старший преподаватель кафедры программного обеспечения  
вычислительной техники и автоматизированных систем*

**Аннотация**

В статье рассматривается система требований к разработке программного продукта R Tutorial. Подробно рассматриваются функциональные и не функциональные требования, применяемые к продуктам данного типа, обосновывается методология разработки и выбор языка реализации.

**Ключевые слова:** язык R, статистическая среда

**Development of the technical task for the software product R Tutorial**

*Dmitrieva Viktorija Aleksandrovna*

*Katanov Khakass State University  
graduate student*

*Grigoreva Sofia Sergeevna*

*Katanov Khakass State University  
graduate student*

*Scientific adviser:*

*Golubnichiy Artem Aleksandrovich*

*Katanov Khakass State University  
senior lecturer department of computing software and automated systems*

**Abstract**

The article describes the system of requirements for the development of the software product R Tutorial. The functional and non-functional requirements applied to products of this type are considered in detail, the development methodology and the language of implementation are substantiated.

**Keywords:** R language, statistical environment

Техническое задание представляет собой исходный документ на проектирование технического объекта или изделия. Программный продукт, являясь также результатом технического творчества, требует составления четкого плана действий для его построения. Разные области познания требуют составления различных технических заданий. Область программной инженерии также не является исключением. Составление как такого технического задания, по сути, тоже относится к видам деятельности, которую необходимо строго регламентировать в соответствии с некоторым планом.

В настоящее время для составления технического задания на наш взгляд целесообразно руководствоваться рядом документов по стандартизации отдельных видов деятельности, принятых в Российской Федерации и разработанных на основе международных стандартов. В области требований к разработке программных продуктов действующими на территории России являются два государственных стандарта: ГОСТ 34.601-90 «Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания» [1] и ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств» [2].

#### ***Сфера применения:***

Программный продукт разрабатывается в целях обучения программированию на языке R, изучению особенностей данного языка и как краткий справочник по данному языку. В программе реализуется также адаптивный тренажер, основными задачами которого, помимо формирования индивидуального пути изучения отдельных тем, также выступает сбор статистической отчетности по изучению отдельных тем языка.

#### ***Функциональные требования:***

Программа должна способствовать процессу обучения пользователей языку программирования R. В программе должны быть реализованы следующие компоненты:

- справочник по особенностям синтаксиса языка R;
- среда для проверки программного кода, реализованная по типу песочницы;
- набор материалов по изучению языка программирования R;
- набор тестовых заданий для проверки закрепления знаний;
- официальная документация языка;
- адаптивный тренажер.

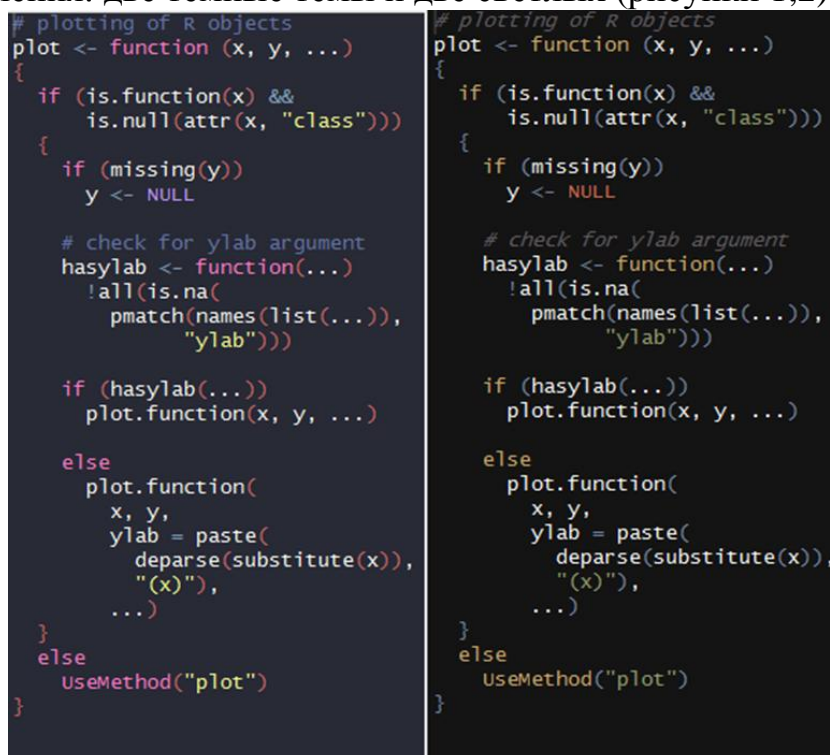
#### ***Требования к программной документации:***

Программная документация должна описывать каждый структурный элемент программы. Назначение структурных элементов их состав и инструкции для работы с ними конечных пользователей.

#### ***Нефункциональные требования:***

#### ***Юзабилити. Требования к интерфейсу:***

Интерфейс программного продукта должен быть выполнен в нейтральных тонах, преимущественно в белом цвете и оттенках серого в градиенте цветов от #ffffff до #999999. Программный продукт также должен иметь возможность смены интерфейса. Должно быть реализовано не менее 4 схем оформления: две темные темы и две светлых (рисунки 1,2).



```
# plotting of R objects
plot <- function(x, y, ...)
{
  if (is.function(x) &&
      is.null(attr(x, "class")))
  {
    if (missing(y))
      y <- NULL

    # check for ylab argument
    hasylab <- function(...)
      !all(is.na(
        pmatch(names(list(...)),
                  "ylab")))

    if (hasylab(...))
      plot.function(x, y, ...)

    else
      plot.function(
        x, y,
        ylab = paste(
          deparse(substitute(x)),
          "(x)",
          ...)
      )
  }
  else
    UseMethod("plot")
}
```

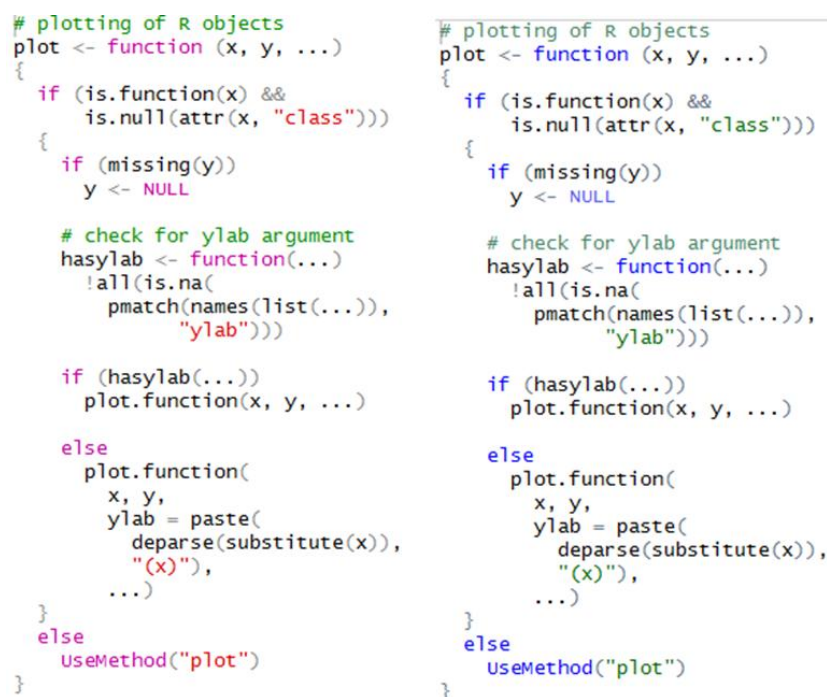
```
# plotting of R objects
plot <- function(x, y, ...)
{
  if (is.function(x) &&
      is.null(attr(x, "class")))
  {
    if (missing(y))
      y <- NULL

    # check for ylab argument
    hasylab <- function(...)
      !all(is.na(
        pmatch(names(list(...)),
                  "ylab")))

    if (hasylab(...))
      plot.function(x, y, ...)

    else
      plot.function(
        x, y,
        ylab = paste(
          deparse(substitute(x)),
          "(x)",
          ...)
      )
  }
  else
    UseMethod("plot")
}
```

Рисунок 1 – Темные схемы оформления (Dracula и Twilight)



```
# plotting of R objects
plot <- function(x, y, ...)
{
  if (is.function(x) &&
      is.null(attr(x, "class")))
  {
    if (missing(y))
      y <- NULL

    # check for ylab argument
    hasylab <- function(...)
      !all(is.na(
        pmatch(names(list(...)),
                  "ylab")))

    if (hasylab(...))
      plot.function(x, y, ...)

    else
      plot.function(
        x, y,
        ylab = paste(
          deparse(substitute(x)),
          "(x)",
          ...)
      )
  }
  else
    UseMethod("plot")
}
```

```
# plotting of R objects
plot <- function(x, y, ...)
{
  if (is.function(x) &&
      is.null(attr(x, "class")))
  {
    if (missing(y))
      y <- NULL

    # check for ylab argument
    hasylab <- function(...)
      !all(is.na(
        pmatch(names(list(...)),
                  "ylab")))

    if (hasylab(...))
      plot.function(x, y, ...)

    else
      plot.function(
        x, y,
        ylab = paste(
          deparse(substitute(x)),
          "(x)",
          ...)
      )
  }
  else
    UseMethod("plot")
}
```

Рисунок 2 – Светлые схемы оформления (Xcode и TextMate)

Программный продукт также должен иметь возможность выбора используемых шрифтов.

Основные структурные компоненты программного продукта должны быть доступны из главного окна программы.

***Надежность:***

При реализации компонентов системы отдельное внимание стоит уделить механизму проверки программного кода языка. При разработке данного модуля необходимо использовать метод «песочница». Модуль должен быть настроен таким образом, что максимально используемый объем памяти не должен превышать 10% от имеющегося в аппаратном обеспечении. Время на исполнение кода должно ограничиваться 30 секундами.

***Производительность:***

Показатели скорости запуска компонентов системы не должны превышать более чем на 30% аналогичные значения при запуске других мобильных приложений для изучения программирования.

***Требования к информационной и программной совместимости:***

Программный продукт должен работать на платформах iOS 9 и выше, на всех поддерживаемых моделях iPhone.

***Методология и стадии разработки:***

Для разработки программного продукта выбрана методология Agile software development.

В рамках технического задания к разработке выделены основные функции, которые должен выполнять программный продукт: справочник по особенностям синтаксиса языка R; среда для проверки программного кода, реализованная по типу песочницы; набор материалов по изучению языка программирования R; набор тестовых заданий для проверки закрепления знаний; официальная документация языка; адаптивный тренажер.

## **Библиографический список**

1. ГОСТ 34.601-90 «Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания»
2. ГОСТ Р ИСО/МЭК 12207-2010 «Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств»