

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

"Московский государственный технический университет информационных технологий, радиотехники и

электроники"

МГТУ МИРЭА

Институт информационных технологий

Кафедра ИТС

Лабораторная работа №1

по дисциплине «Программная организация

высокопроизводительных вычислений»

Преподаватель:

Торхов Алексей Евгеньевич

Исполнитель:

Пыжов Владислав Олегович

Группа: ИКМО-2-15

Москва 2016

# Постановка задачи

Цель данной лабораторной работы состоит в применении основных аспектов современного подхода к разработке программного обеспечению, таким как использование систем контроля версий, тестирование, статический анализ кода и применение стандартов оформления кода, к решению практической задачи.

Для её достижения необходимо выполнить следующие задачи:

1. Установить систему контроля версий (Git / SVN и т.д.) и создать репозиторий, который впоследствии будет наполняться результатами выполнения задач данной лабораторной работы и последующих;

2. Разработать пагинатор с функцией, реализующей вывод соответствующего ему html-кода;

3. Написать модульные тесты для пагинатора и сделать оценку покрытия тестами приложения.

4. Оформить код в соответствии руководством по оформлению кода и провести статический анализ кода.

# 1. Настройка работы с системой контроля версий

## Создание репозитория и размещение его на хостинге Github

Git – распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года.

Структура репозитория имеет следующий вид:

|  |
| --- |
| **Exercies** |
| **labs** |
| **FirstLab** |
| **SecondLab** |
| **ThirdLab** |
| **FourthLab** |
| .gitignore |
| README.md |

Рисунок 1.1 – Структура репозитория

После того, как репозиторий был создан, он был размещен на бесплатном для проектов с открытым исходным кодом сервере GitHub по адресу:

<https://github.com/Git-Lord/Exercies>

Он представлен на следующем скриншоте:

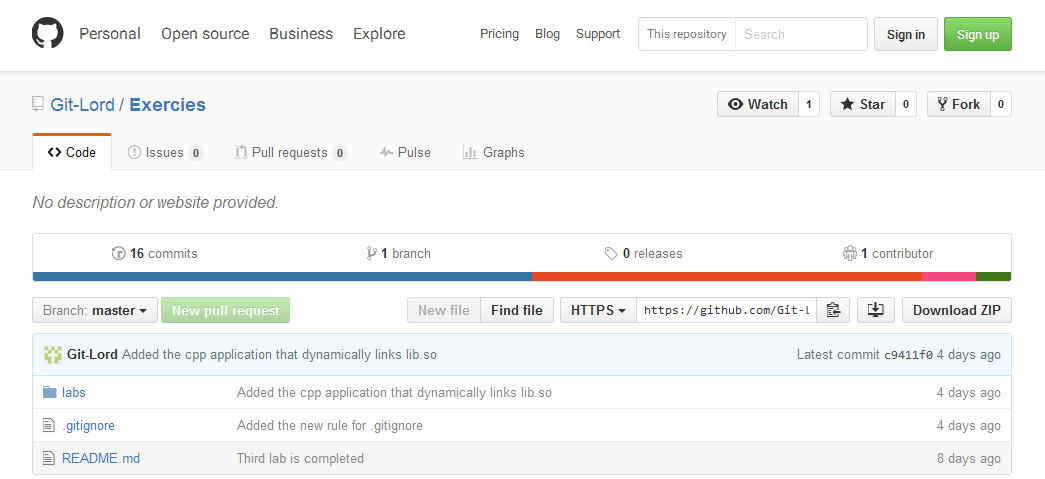


Рисунок 1.2 – Репозиторий на GitHub

# 2. Разработка пагинатора

## Программный код

Для написания пагинатора был выбран язык высокого уровня Python. Сам он оформлен в виде класса, который имеет методы, отвечающие за логику (конструктор, generate\_link, next\_page, previous\_page, create\_page, create\_divider, create\_paginator) и метод, отвечающий за представление, т.е. генерацию html кода (\_\_str\_\_).

|  |
| --- |
| # -\*- coding: utf-8 -\*-  import math  class Paginator:  """ The main class, that creates a paginator: """  def \_\_init\_\_(self, current\_page, count\_pages, max\_pages, pattern\_url=""):  if current\_page > count\_pages or current\_page < 1:  raise Exception("The incorrect page number")  self.current\_page = current\_page  self.count\_pages = count\_pages  self.pattern\_url = pattern\_url  self.max\_pages = max\_pages  def generate\_link(self, number):  return self.pattern\_url.replace("{{number}}", str(number))  def previous\_page(self):  if self.current\_page > 1:  return self.current\_page - 1  else:  return None  def next\_page(self):  if self.current\_page > self.count\_pages:  return None  else:  return self.current\_page + 1  def create\_page(self, number):  return {"number": number, "link": self.generate\_link(number),  "active": self.current\_page == number}  def create\_divider(self):  return {"number": "...", "link": None, "active": False}  def create\_paginator(self):  """ Generates an array of pages like the following an array:  [  {"number": 1, "link": "/page1/", "active": False},  {"number": "...", "link": None, "active": False},  {"number": 3, "link": "/page3/", "active": False},  {"number": 4, "link": "/page4/", "active": True},  {"number": 5, "link": "/page5/", "active": False},  {"number": "...", "link": None, "active": False},  {"number": 10, "link": "/page10/", "active": False}  ]  """  result = []  number\_pages = min(self.count\_pages, self.max\_pages)  left\_trail = int(math.floor(number\_pages/2))  right\_trail = int(math.floor(number\_pages/2))  if not number\_pages % 2:  left\_trail -= 1  left\_shift = self.current\_page - left\_trail  if left\_shift <= 0:  right\_trail -= left\_shift - 1  left\_trail += left\_shift  right\_shift = self.current\_page + right\_trail  if right\_shift > self.count\_pages:  right\_trail -= right\_shift - self.count\_pages  left\_trail += right\_shift - self.count\_pages  if self.count\_pages > 3:  start = self.current\_page - left\_trail  end = self.current\_page + right\_trail  if end != self.count\_pages:  end -= 1  if start != 1:  start += 1  if start != 1:  result.append(self.create\_page(1))  if start > 2:  result.append(self.create\_divider())  for i in range(start, end+1):  result.append(self.create\_page(i))  if end < self.count\_pages - 1:  result.append(self.create\_divider())  if end != self.count\_pages:  result.append(self.create\_page(self.count\_pages))  else:  for i in range(0, self.count\_pages):  result.append(self.create\_page(i+1))  return result  def \_\_str\_\_(self):  """ Generates a string of html code like the following string  <ul class = "pagination">  <li><a href="/page3/">←</a></li>  <li><a href="/page1/">1</a></li>  <li class="disabled"><a href="#">...</a></li>  <li><a href="/page3/">3</a></li>  <li class="active"><a href="/page4/">4</a></li>  <li><a href="/page5/">5</a></li>  <li class="disabled"><a href="#">...</a></li>  <li><a href="/page10/">10</a></li>  <li><a href="/page5/">→</a></li>  </ul>  """  # Creates a array, that describes the paginator  pages = self.create\_paginator()  # Creates the string of a html code from that array  html = '<ul class = "pagination">\n{0}</ul> \n'  ul = ""  if self.previous\_page() is not None:  ul += '\t\t<li><a href="{0}">←</a></li>\n'.format(self.generate\_link(self.previous\_page()))  for page in pages:  if page["link"] is not None:  if page["active"]:  parameters = ' class="active"'  else:  parameters = ''  href = page["link"]  else:  parameters = ' class="disabled"'  href = "#"  ul += '\t\t<li{1}><a href="{0}">{2}</a></li>\n'.format(href, parameters, str(page["number"]))  if self.next\_page() is not None:  ul += '\t\t<li><a href="{0}">→</a></li>\n'.format(self.generate\_link(self.next\_page()))  return html.format(ul) |

## Демонстрация работы

Далее представлены примеры работы пагинатора, т.е. сгенерированный им HTML-код, для их стилизации была применена каскадная таблица стилей от Bootstrap.

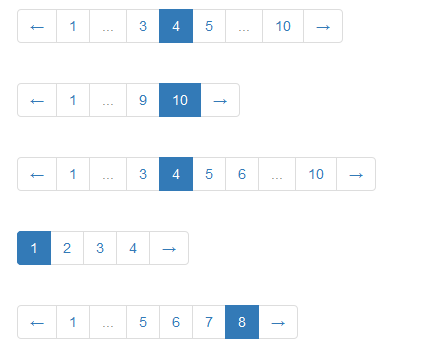


Рисунок 2.1 – Пример работы программы

# 3. Тестирование программного кода

## Исходные коды тестов

Для тестирования был выбран стандартный инструмент для Python: модуль unittest.

Исходный код тест-кейсов выглядит следующим образом:

|  |
| --- |
| # -\*- coding: utf-8 -\*-  import unittest  from tests\_cases import \*  class PaginatorTest(unittest.TestCase):  # Test a generation of paginator:  def test\_generate\_page(self):  for page in generated\_pages:  self.assertEqual(page["result"].create\_paginator(), page["expected"])  # Test a generation of html:  def test\_generate\_html(self):  for html in generated\_html:  self.assertEqual(str(html["result"]).replace("\t", "").replace("\n", ""), html["expected"])    # Test a length of paginator:  def test\_length(self):  for count\_pages in range(1,100):  for max\_pages in range(3,count\_pages+1):  for current\_page in range(1,count\_pages+1):  paginator = Paginator(current\_page=current\_page,  count\_pages=count\_pages, pattern\_url="/page{{number}}/",  max\_pages = max\_pages)  pages = paginator.create\_paginator()  count = 0  for page in pages:  if page["link"] is not None:  count += 1  self.assertEqual(count, max\_pages)  # Save a html file:  def test\_save\_html(self):  content = '<html>'\  '\n<head>'\  '\n\t<link rel="stylesheet" href="bootstrap.min.css">\n\t<meta charset="UTF-8">' \  '\n</head>' \  '\n<body>\n{0}\n</body>\n' \  '</html>'  body = ""  for example in generated\_examples:  body += "<div> {0} </div>\n".format(str(example))  file = open("html/main.html", "w", encoding="utf8")  file.write(content.format(body))  file.close()  if \_\_name\_\_ == '\_\_main\_\_':  unittest.main() |

Набор данных для теста, описан в другом модуле:

|  |
| --- |
| # -\*- coding: utf-8 -\*-  from main import Paginator  generated\_examples = [  Paginator(current\_page=4, count\_pages=10, pattern\_url="/page{{number}}/", max\_pages = 5),  Paginator(current\_page=10, count\_pages=10, pattern\_url="/page{{number}}/", max\_pages = 3),  Paginator(current\_page=4, count\_pages=10, pattern\_url="/page{{number}}/", max\_pages = 6),  Paginator(current\_page=1, max\_pages = 10, count\_pages=4, pattern\_url="/page\_{{number}}/"),  Paginator(current\_page=8, max\_pages = 5, count\_pages=8, pattern\_url="/page\_{{number}}/"),  Paginator(current\_page=41, max\_pages = 10, count\_pages=50, pattern\_url="/page\_{{number}}/"),  ]  generated\_html = [  {  "result": Paginator(current\_page=4, count\_pages=10, pattern\_url="/page{{number}}/", max\_pages = 5),  "expected": '<ul class = "pagination">'  '<li><a href="/page3/">←</a></li>'  '<li><a href="/page1/">1</a></li>'  '<li class="disabled"><a href="#">...</a></li>'  '<li><a href="/page3/">3</a></li>'  '<li class="active"><a href="/page4/">4</a></li>'  '<li><a href="/page5/">5</a></li>'  '<li class="disabled"><a href="#">...</a></li>'  '<li><a href="/page10/">10</a></li>'  '<li><a href="/page5/">→</a></li>'  '</ul> '  }  ]  generated\_pages = [  {  "result": Paginator(current\_page=4, count\_pages=10, pattern\_url="/page{{number}}/", max\_pages = 5),  "expected":  [  {"number": 1, "link": "/page1/", "active": False},  {"number": "...", "link": None, "active": False},  {"number": 3, "link": "/page3/", "active": False},  {"number": 4, "link": "/page4/", "active": True},  {"number": 5, "link": "/page5/", "active": False},  {"number": "...", "link": None, "active": False},  {"number": 10, "link": "/page10/", "active": False}  ]  },  {  "result": Paginator(current\_page=1, count\_pages=10,  pattern\_url="/page{{number}}/", max\_pages = 3),  "expected": [  {"number": 1, "link": "/page1/", "active": True},  {"number": 2, "link": "/page2/", "active": False},  {"number": "...", "link": None, "active": False},  {"number": 10, "link": "/page10/", "active": False},  ]  },  {  "result": Paginator(current\_page=10, count\_pages=10,  pattern\_url="/page{{number}}/", max\_pages = 3),  "expected": [  {"number": 1, "link": "/page1/", "active": False},  {"number": "...", "link": None, "active": False},  {"number": 9, "link": "/page9/", "active": False},  {"number": 10, "link": "/page10/", "active": True},  ]  },  {  "result": Paginator(current\_page=2, count\_pages=10,  pattern\_url="/page{{number}}/", max\_pages = 4),  "expected": [  {"number": 1, "link": "/page1/", "active": False},  {"number": 2, "link": "/page2/", "active": True},  {"number": 3, "link": "/page3/", "active": False},  {"number": "...", "link": None, "active": False},  {"number": 10, "link": "/page10/", "active": False},  ]  },  {  "result": Paginator(current\_page=3, count\_pages=5,  pattern\_url="/page{{number}}/", max\_pages = 5),  "expected": [  {"number": 1, "link": "/page1/", "active": False},  {"number": 2, "link": "/page2/", "active": False},  {"number": 3, "link": "/page3/", "active": True},  {"number": 4, "link": "/page4/", "active": False},  {"number": 5, "link": "/page5/", "active": False},  ]  },  ] |

## Оценка покрытия тестами

Для оценки покрытия тестами используется модуль coverage. Результат анализа, выполненный им, представлен далее:

|  |
| --- |
| Name Stmts Miss Cover  ------------------------------------  main.py 56 2 96%  tests.py 19 0 100%  tests\_cases.py 4 0 100%  ------------------------------------  TOTAL 79 2 97% |

# 

# 4. Качество кода

## Статический анализ кода

Для статического анализа кода используется модуль PyLint, результат анализа которого представлен далее:

|  |
| --- |
| \*\*\*\*\*\*\*\*\*\*\*\*\* Module main  C: 88, 0: Line too long (103/100) (line-too-long)  C: 99, 0: Line too long (105/100) (line-too-long)  C: 1, 0: Missing module docstring (missing-docstring)  C: 16, 4: Missing method docstring (missing-docstring)  C: 19, 4: Missing method docstring (missing-docstring)  C: 25, 4: Missing method docstring (missing-docstring)  C: 31, 4: Missing method docstring (missing-docstring)  C: 35, 4: Missing method docstring (missing-docstring)  R: 35, 4: Method could be a function (no-self-use)  C: 86, 8: Invalid variable name "ul" (invalid-name)  C: 88,12: Invalid variable name "ul" (invalid-name)  C: 99,12: Invalid variable name "ul" (invalid-name)  C:101,12: Invalid variable name "ul" (invalid-name)  Report  ======  56 statements analysed.  Statistics by type  ------------------  +---------+-------+-----------+-----------+------------+---------+  |type |number |old number |difference |%documented |%badname |  +=========+=======+===========+===========+============+=========+  |module |1 |NC |NC |0.00 |0.00 |  +---------+-------+-----------+-----------+------------+---------+  |class |1 |NC |NC |100.00 |0.00 |  +---------+-------+-----------+-----------+------------+---------+  |method |8 |NC |NC |37.50 |0.00 |  +---------+-------+-----------+-----------+------------+---------+  |function |0 |NC |NC |0 |0 |  +---------+-------+-----------+-----------+------------+---------+  Raw metrics  -----------  +----------+-------+------+---------+-----------+  |type |number |% |previous |difference |  +==========+=======+======+=========+===========+  |code |62 |59.62 |NC |NC |  +----------+-------+------+---------+-----------+  |docstring |26 |25.00 |NC |NC |  +----------+-------+------+---------+-----------+  |comment |5 |4.81 |NC |NC |  +----------+-------+------+---------+-----------+  |empty |11 |10.58 |NC |NC |  +----------+-------+------+---------+-----------+  Duplication  -----------  +-------------------------+------+---------+-----------+  | |now |previous |difference |  +=========================+======+=========+===========+  |nb duplicated lines |0 |NC |NC |  +-------------------------+------+---------+-----------+  |percent duplicated lines |0.000 |NC |NC |  +-------------------------+------+---------+-----------+  Messages by category  --------------------  +-----------+-------+---------+-----------+  |type |number |previous |difference |  +===========+=======+=========+===========+  |convention |12 |NC |NC |  +-----------+-------+---------+-----------+  |refactor |1 |NC |NC |  +-----------+-------+---------+-----------+  |warning |0 |NC |NC |  +-----------+-------+---------+-----------+  |error |0 |NC |NC |  +-----------+-------+---------+-----------+  Messages  --------  +------------------+------------+  |message id |occurrences |  +==================+============+  |missing-docstring |6 |  +------------------+------------+  |invalid-name |4 |  +------------------+------------+  |line-too-long |2 |  +------------------+------------+  |no-self-use |1 |  +------------------+------------+  Global evaluation  -----------------  Your code has been rated at 7.68/10 |