# **3.3 Проектування алгоритмів роботи методів програмних класів**

Алгоритм методу Registration(User user) зображений на рисунку 3.3.1.

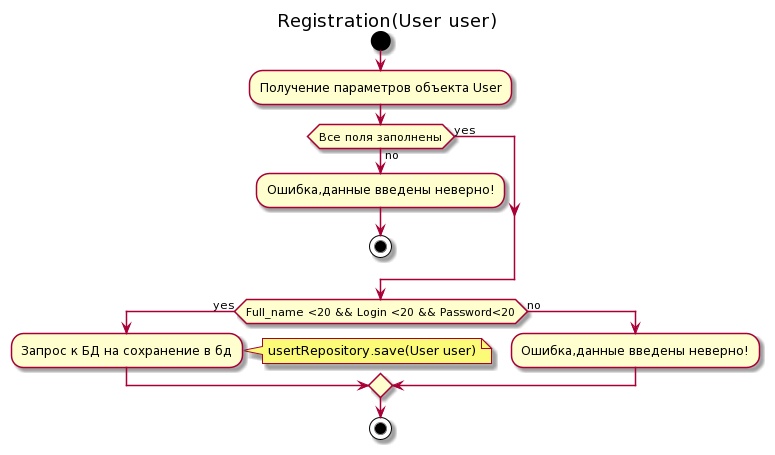


Рисунок 3.3.1 – Алгоритм методу Registration(User user)

Код до методу Registration(User user):

@startuml

start

title Registration(User user)

:Получение параметров объекта User;

if (Все поля заполнены) then (yes)

else (no)

:Ошибка,данные введены неверно!;

stop

endif

if (Full\_name <20 && Login <20 && Password<20) then (yes)

:Запрос к БД на сохранение в бд;

note right

usertRepository.save(User user)

end note

else (no)

:Ошибка,данные введены неверно!;

endif

stop

@enduml

Алгоритм методу Authentication(string Login, string password) зображений на рисунку 3.3.2.

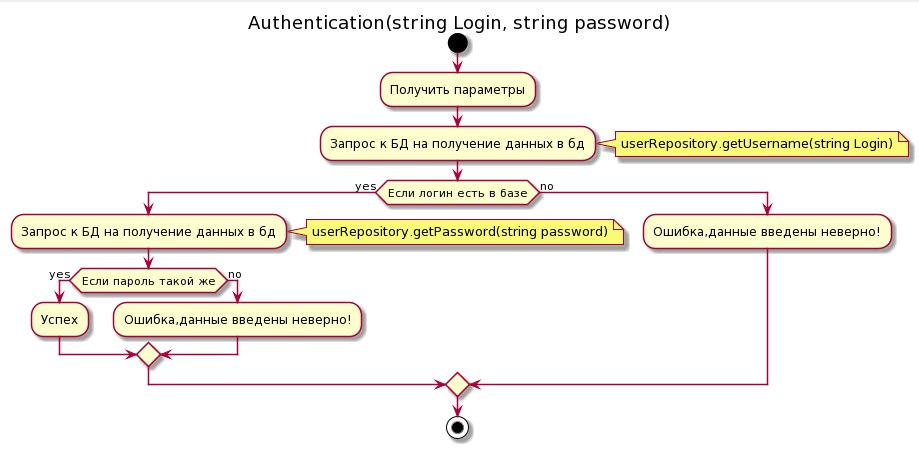


Рисунок 3.3.2 – Алгоритм методу Authentication(User user)

Код до методу Authentication(string Login, string password):

@startuml

start

title Authentication(string Login, string password)

:Получить параметры;

:Запрос к БД на получение данных в бд;

note right

userRepository.getUsername(string Login)

end note

if (Если логин есть в базе) then (yes)

:Запрос к БД на получение данных в бд;

note right

userRepository.getPassword(string password)

end note

if (Если пароль такой же) then (yes)

:Успех;

else (no)

:Ошибка,данные введены неверно!;

endif

else (no)

:Ошибка,данные введены неверно!;

endif

stop

@enduml

Алгоритм методу getEvent()зображений на рисунку 3.3.3.



Рисунок 3.3.3 – Алгоритм методу Registration(User user)

Код до методу getEvent():

@startuml

start

title getEvent()

:Получение через параметры username;

:Запрос к БД на получение всех событий пользователя;

note right

EventRepository.findForUsername(string username)

end note

:Вернуть все события;

stop

@enduml

Алгоритм методу getTask() зображений на рисунку 3.3.4.

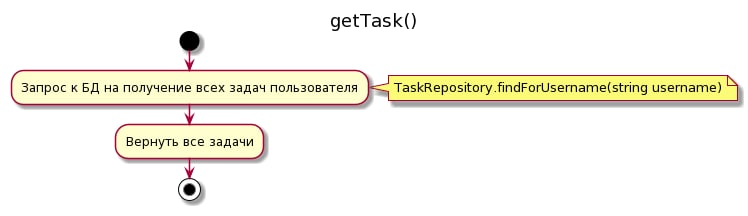


Рисунок 3.3.4 – Алгоритм методу getTask()

Код до методу getTask():

@startuml

start

title getTask()

:Запрос к БД на получение всех задач пользователя;

note right

TaskRepository.findForUsername(string username)

end note

:Вернуть все задачи;

stop

@enduml

Алгоритм методу getNote() зображений на рисунку 3.3.5.

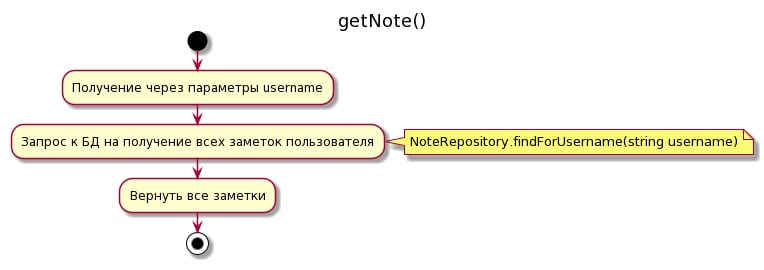


Рисунок 3.3.5 – Алгоритм методу getNote()

Код до методу Registration(User user):

@startuml

start

title getNote()

:Получение через параметры username;

:Запрос к БД на получение всех заметок пользователя;

note right

NoteRepository.findForUsername(string username)

end note

:Вернуть все заметки;

stop

@enduml

Алгоритм методу getRecommendation() зображений на рисунку 3.3.6.



Рисунок 3.3.6 – Алгоритм методу getRecommendation()

Код до методу getRecommendation():

@startuml

start

title getRecommendation()

:Получение через параметры username;

:Запрос к БД на получение всех рекомендации для пользователя;

note right

RecommendationRepository.findForUsername(string username)

end note

:Вернуть все рекомендации;

stop

@enduml

Алгоритм методу getPhone() зображений на рисунку 3.3.7.

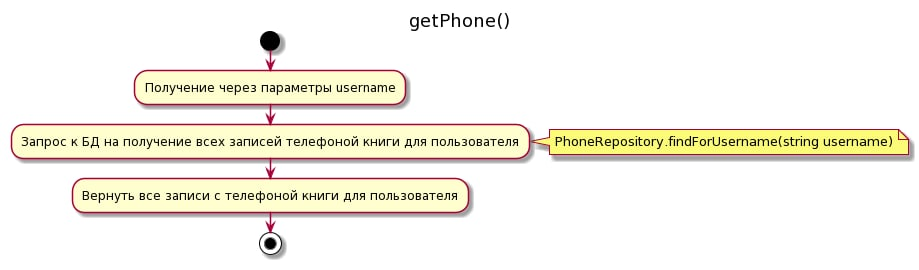


Рисунок 3.3.7 – Алгоритм методу getPhone()

Код до методу getPhone():

@startuml

start

title getPhone()

:Получение через параметры username;

:Запрос к БД на получение всех записей телефоной книги для пользователя;

note right

PhoneRepository.findForUsername(string username)

end note

:Вернуть все записи с телефоной книги для пользователя;

stop

@endumlelse (no)

:Ошибка,данные введены неверно!;

endif

stop

Алгоритм методу getDiary() зображений на рисунку 3.3.8.



Рисунок 3.3.8 – Алгоритм методу getDiary()

Код до методу getDiary():

@startuml

start

title getDiary()

:Получение через параметры username;

:Запрос к БД на получение всех личных записей пользователя;

note right

DiaryRepository.findForUsername(string username)

end note

:Вернуть все личные записи пользователя;

stop

@enduml

Алгоритм методу getPasswordManadger() зображений на рисунку 3.3.9.

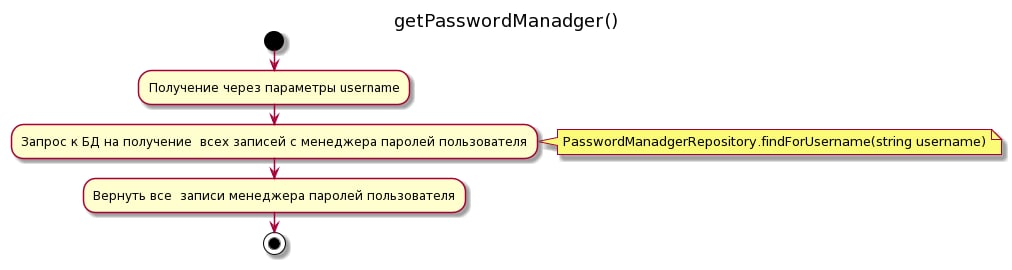


Рисунок 3.3.9 – Алгоритм методу getPasswordManadger()

Код до методу getPasswordManadger():

@startuml

start

title getPasswordManadger()

:Получение через параметры username;

:Запрос к БД на получение всех записей с менеджера паролей пользователя;

note right

PasswordManadgerRepository.findForUsername(string username)

end note

:Вернуть все записи менеджера паролей пользователя;

stop

@enduml

Алгоритм методу setTask(Task task) зображений на рисунку 3.3.10.

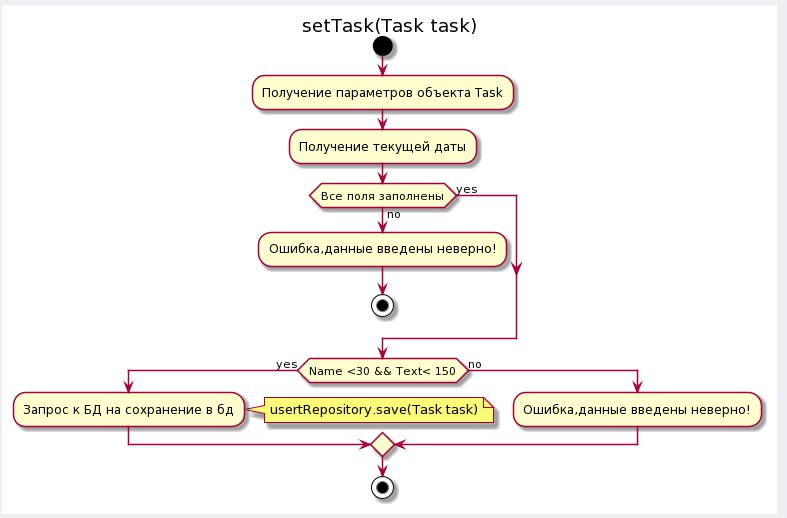


Рисунок 3.3.10 – Алгоритм методу setTask(Task task)

Код до методу setTask(Task task):

@startuml

start

title setTask(Task task)

:Получение параметров объекта Task;

:Получение текущей даты;

if (Все поля заполнены) then (yes)

else (no)

:Ошибка,данные введены неверно!;

stop

endif

if (Name <30 && Text< 150) then (yes)

:Запрос к БД на сохранение в бд;

note right

usertRepository.save(Task task)

end note

else (no)

:Ошибка,данные введены неверно!;

endif

stop

@enduml

Алгоритм методу setNote(Note note) зображений на рисунку 3.3.11.

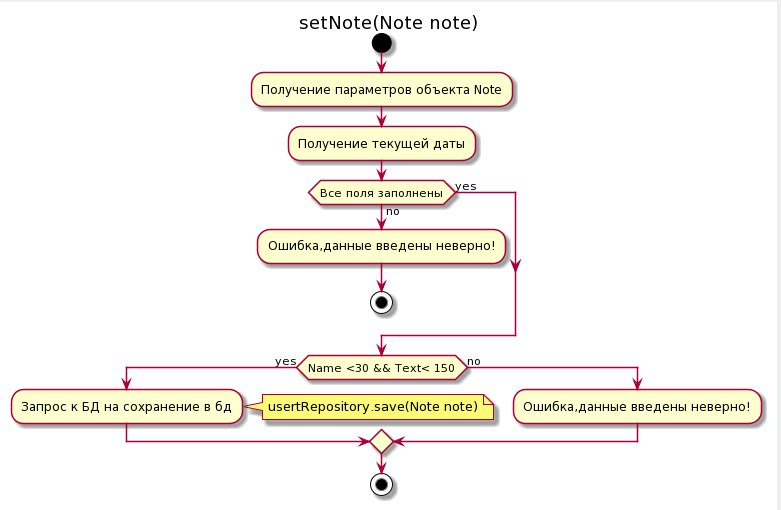


Рисунок 3.3.11 – Алгоритм методу setNote(Note note)

Код до методу setNote(Note note):

@startuml

start

title setNote(Note note)

:Получение параметров объекта Note;

:Получение текущей даты;

if (Все поля заполнены) then (yes)

else (no)

:Ошибка,данные введены неверно!;

stop

endif

if (Name <30 && Text< 150) then (yes)

:Запрос к БД на сохранение в бд;

note right

usertRepository.save(Note note)

end note

else (no)

:Ошибка,данные введены неверно!;

endif

stop @enduml

Алгоритм методу setEvent(Event event) зображений на рисунку 3.3.12.

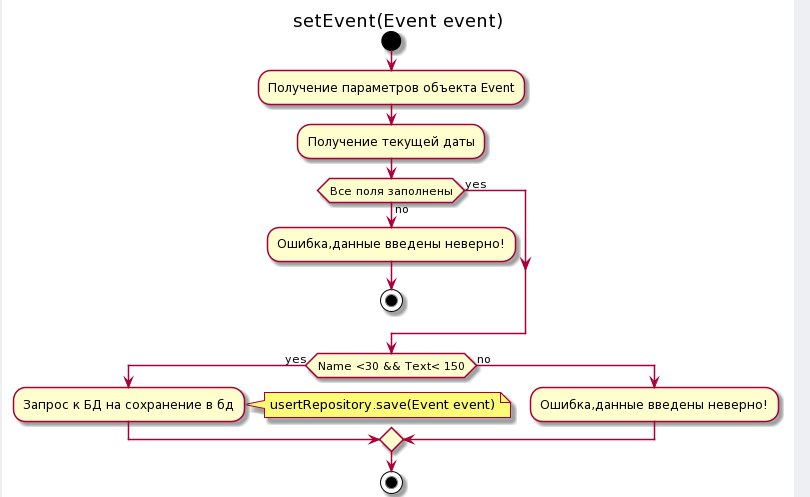


Рисунок 3.3.12 – Алгоритм методу setEvent(Event event)

Код до методу setEvent(Event event):

@startuml

start

title setEvent(Event event)

:Получение параметров объекта Event;

:Получение текущей даты;

if (Все поля заполнены) then (yes)

else (no)

:Ошибка,данные введены неверно!;

stop

endif

if (Name <30 && Text< 150) then (yes)

:Запрос к БД на сохранение в бд;

note right

usertRepository.save(Event event)

end note

else (no)

:Ошибка,данные введены неверно!;

endif

stop

@enduml

Алгоритм методу setRecommendation(Recommendation recommendation) зображений на рисунку 3.3.13.

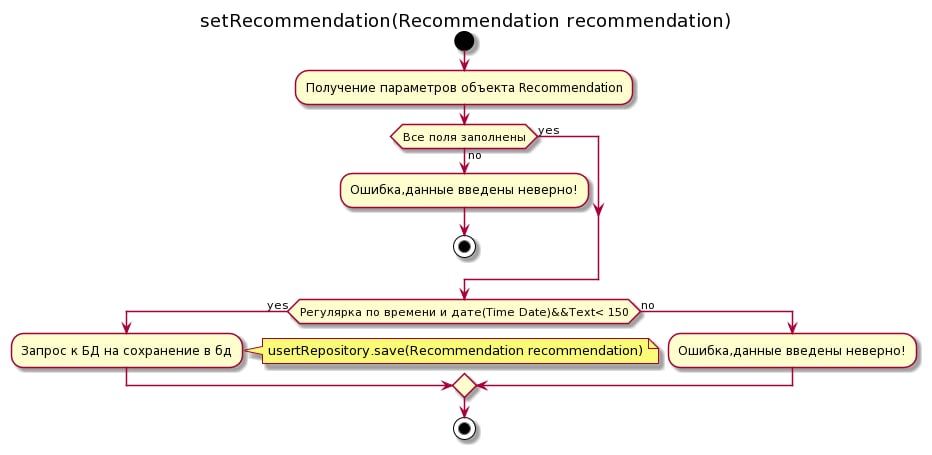


Рисунок 3.3.13 – Алгоритм методу setRecommendation(Recommendation recommendation)

Код до методу setRecommendation(Recommendation recommendation):

@startuml

start

title setRecommendation(Recommendation recommendation)

:Получение параметров объекта Recommendation;

if (Все поля заполнены) then (yes)

else (no)

:Ошибка,данные введены неверно!;

stop

endif

if (Регулярка по времени и дате(Time Date)&&Text< 150) then (yes)

:Запрос к БД на сохранение в бд;

note right

usertRepository.save(Recommendation recommendation)

end note

else (no)

:Ошибка,данные введены неверно!;

endif

stop

@enduml

Алгоритм методу setPhone(PhoneBook phoneBook) зображений на рисунку 3.3.14.

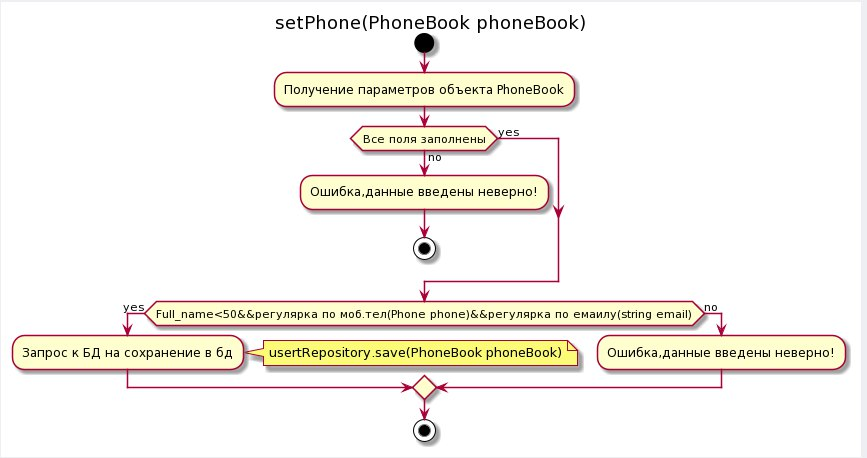


Рисунок 3.3.14 – Алгоритм методу setPhone(PhoneBook phoneBook)

Код до метод setPhone(PhoneBook phoneBook):

@startuml

start

title setPhone(PhoneBook phoneBook)

:Получение параметров объекта PhoneBook;

if (Все поля заполнены) then (yes)

else (no)

:Ошибка,данные введены неверно!;

stop

endif

if (Full\_name<50&&регулярка по моб.тел(Phone phone)&&регулярка по емаилу(string email)) then (yes)

:Запрос к БД на сохранение в бд;

note right

usertRepository.save(PhoneBook phoneBook)

end note

else (no)

:Ошибка,данные введены неверно!;

endif

stop

@enduml

Алгоритм методу setPersonalDiary(PersonalDiary personalDiary) зображений на рисунку 3.3.15.

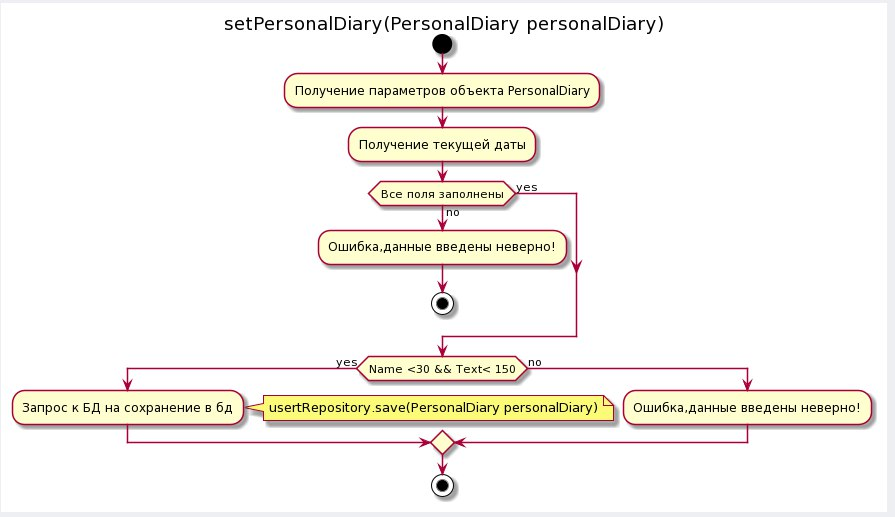


Рисунок 3.3.15 – Алгоритм методу setPersonalDiary(PersonalDiary personalDiary)

Код до методу setPersonalDiary(PersonalDiary personalDiary):

@startuml

start

title setPersonalDiary(PersonalDiary personalDiary)

:Получение параметров объекта PersonalDiary;

:Получение текущей даты;

if (Все поля заполнены) then (yes)

else (no)

:Ошибка,данные введены неверно!;

stop

endif

if (Name <30 && Text< 150) then (yes)

:Запрос к БД на сохранение в бд;

note right

usertRepository.save(PersonalDiary personalDiary)

end note

else (no)

:Ошибка,данные введены неверно!;

endif

stop@enduml@enduml

Алгоритм методу setPasswordManager(PasswordManager passwordManager) зображений на рисунку 3.3.16.

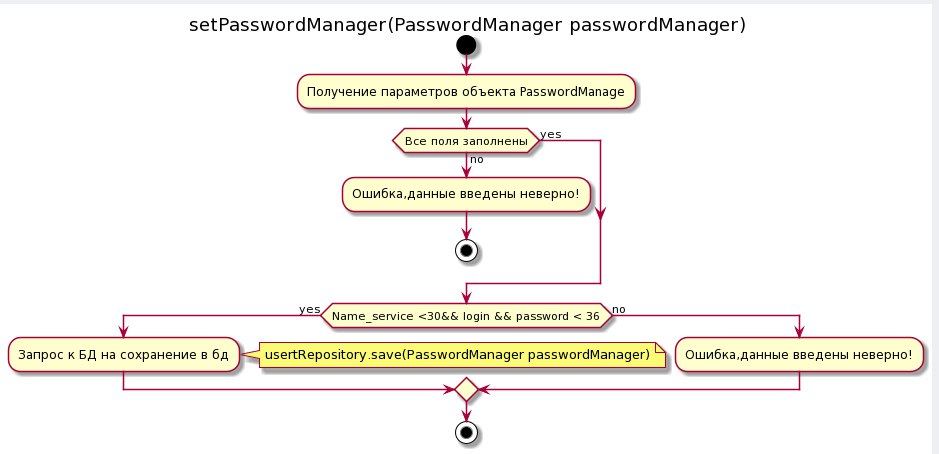


Рисунок 3.3.16 – Алгоритм методу setPasswordManager(PasswordManager passwordManager)

Код до методу setPasswordManager(PasswordManager passwordManager):

@startuml

start

title setPasswordManager(PasswordManager passwordManager)

:Получение параметров объекта PasswordManage;

if (Все поля заполнены) then (yes)

else (no)

:Ошибка,данные введены неверно!;

stop

endif

if (Name\_service <30&& login && password < 36) then (yes)

:Запрос к БД на сохранение в бд;

note right

usertRepository.save(PasswordManager passwordManager)

end note

else (no)

:Ошибка,данные введены неверно!;

endif

stop

@enduml

Алгоритм методу DeleteEvent(int id)зображений на рисунку 3.3.17.

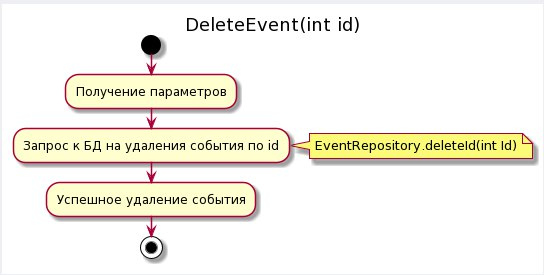


Рисунок 3.3.17 – Алгоритм методу DeleteEvent(int id)

Код до методу DeleteEvent(int id):

@startuml

start

title DeleteEvent(int id)

:Получение параметров;

:Запрос к БД на удаления события по id;

note right

EventRepository.deleteId(int Id)

end note

:Успешное удаление события;

stop

@enduml

Алгоритм методу DeleteTask(int id) зображений на рисунку 3.3.18.



Рисунок 3.3.18 – Алгоритм методу DeleteTask(int id)

Код до методу DeleteTask(int id):

@startuml

start

title DeleteTask(int id)

:Получение параметров;

:Запрос к БД на удаления задачи по id;

note right

TaskRepository.deleteId(int Id)

end note

:Успешное удаление;

stop

@enduml

Алгоритм методу DeleteNote(int id) зображений на рисунку 3.3.19.

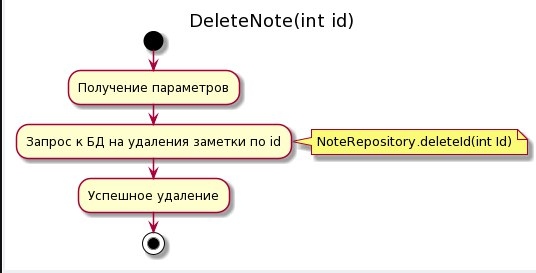


Рисунок 3.3.19 – Алгоритм методу DeleteNote(int id)

Код до методу DeleteNote(int id):

@startuml

start

title DeleteNote(int id)

:Получение параметров;

:Запрос к БД на удаления заметки по id;

note right

NoteRepository.deleteId(int Id)

end note

:Успешное удаление;

Stop @enduml@enduml

Алгоритм методу DeleteRecommendation (int id) зображений на рисунку 3.3.20.

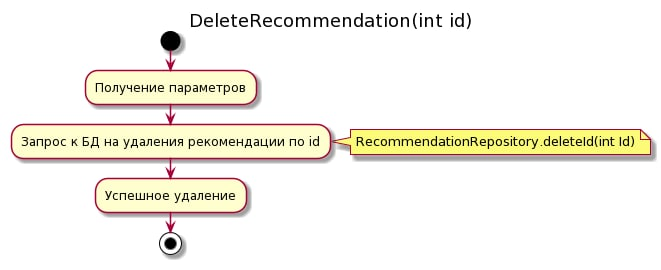


Рисунок 3.3.20 – Алгоритм методу DeleteRecommendation (int id)

Код до методу DeleteRecommendation (int id):

@startuml

start

title DeleteRecommendation(int id)

:Получение параметров;

:Запрос к БД на удаления рекомендации по id;

note right

RecommendationRepository.deleteId(int Id)

end note

:Успешное удаление;

stop

@enduml

Алгоритм методу DeletePasswordManager(int id)зображений на рисунку 3.3.21.

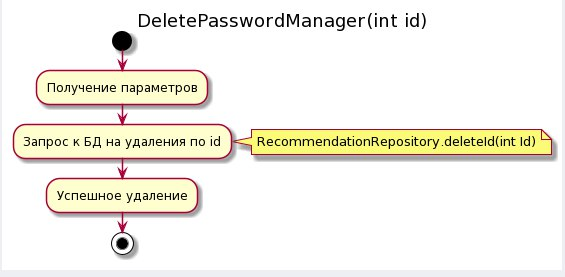


Рисунок 3.3.21 – Алгоритм методу DeletePasswordManager(int id)

Код до методу DeletePasswordManager(int id):

@startuml

start

title DeletePasswordManager(int id)

:Получение параметров;

:Запрос к БД на удаления по id;

note right

RecommendationRepository.deleteId(int Id)

end note

:Успешное удаление;

stop

@enduml

Алгоритм методу DeletePersonalDiary (int id)зображений на рисунку 3.3.22.



Рисунок 3.3.22 – Алгоритм методу DeletePersonalDiary (int id)

Код до методу DeletePersonalDiary (int id):

@startuml

start

title DeletePersonalDiary(int id)

:Получение параметров;

:Запрос к БД на удаления по id;

note right

DeletePersonalDiaryRepository.deleteId(int Id)

end note :Успешное удаление; stop @enduml

Алгоритм методу DeletePasswordManager(int id)зображений на рисунку 3.3.23.



Рисунок 3.3.23 – Алгоритм методу DeletePasswordManager(int id)

Код до методу DeletePasswordManager(int id):

@startuml

start

title PhoneBook(int id)

:Получение параметров;

:Запрос к БД на удаления по id;

note right

phoneBookRepository.deleteId(int Id)

end note

:Успешное удаление;

stop

@enduml